

Dossier de Projet

Raimbault Diane

SANIOS

Sommaire

	2
Introduction	3
Compétences couvertes par le projet	4
Présentation du projet Sanios	4
Conditions du stage	5
GitHub	5
Cahier des charges	6
Lexique	6
Contraintes Globales	6
Spécificités Techniques	9
Maquettage	9
Responsif	12
Sécurité et protection des formulaires.	16
Développer une interface utilisateur web dynamique	18
Définir une base de données cohérente	20
Développer les composants d'accès aux données	23
Interconnexion du code	25
Le session_abort()	27
Création de QR Code	32
CONCLUSION	35

Annexes	36
CAHIER DES CHARGES	37
DÉFINITION DE SANIOS	38
ENJEUX ET OBJECTIFS	38
SPÉCIFICITÉS TECHNIQUES	39
CHARTES GRAPHIQUES	39
APERÇUS DES CONTENUS	40
DIAGRAMMES D'ACTIVITÉ	47
VOCABULAIRE DES PROFILS :	50
CONCEPTION DE LA BASE DE DONNÉES	50
Structure des tables	50
Vues Relationnelles.	57
CONTRAINTES PAR PROFILS	58
VISUALISATION DU CRUD PAR PROFIL	61
DÉMONSTRATION DES FORMULAIRES	62

Introduction

Spécialiste régional du transport sanitaire, l'entreprise Etoile Secours (Groupe Philippon) est un poids moyen du secteur en Charente Maritime. Fort d'une soixantaine d'entreprise, environ 1000 salariés et 500 véhicules, ses services s'articulent autour de plusieurs pôles :

- . Le transport sanitaire des malades et blessés.
- . Les services de pompes funèbres

Dans une optique d'expansion, le groupe à aujourd'hui des besoins informatiques et numériques forts.

Cette multiplicité de véhicules soumis à des inspections strictes de l'ARS (Agence régionale de la santé) à conduit à repenser une application web existante mais dépassée par les nouveaux besoins d'une entreprise en croissance constante : Sanios.

Feuille de route des nettoyages et entretiens d'un véhicule, Sanios devait être simple d'utilisation et visuellement sobre. Sa refonte devait nécessairement reprendre ces contraintes mais également implémenter de nouvelles fonctionnalités et faire évoluer le projet afin de correspondre aux attentes du groupe.

C'est de ce constat qu'est né la volonté d'un projet conjoint qui m'a été attribué lors du stage de fin d'étude en développement web et web mobile.

Il était important pour moi de m'approprier un projet qui me permettait à la fois de continuer à stimuler ma curiosité mais également à asseoir une maîtrise de projet de la création de son cahier des charges, sa conception jusqu'à sa livraison.

Compétences couvertes par le projet

Liste des compétences classées par activité acquises lors du développement du projet.

Activité n°1 : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

- Maquetter une application
- Réaliser une interface utilisateur web statique et adaptable
- Développer une interface utilisateur web dynamique

Activité n°2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile

Liste conforme au REAC (Référentiel Emploi Activités Compétences) du titre Développeur Web et Web Mobile de niveau III .

Présentation du projet Sanios

Afin de comprendre la genèse des besoins de Sanios, il est important de rappeler que le projet est basé sur une application qui existait déjà mais était devenue obsolète. Aussi, le cahier des charges du projet ainsi que ses contraintes étaient en amont définies et rentraient dans une logique de refonte et d'amélioration de l'application web.

L'objectif de cette nouvelle version était de réimplémenter avec un visuel plus actuel des fonctionnalités présentes mais également d'ajouter de nouvelles fonctionnalités pour permettre un meilleur suivi des véhicules de l'entreprise.

En lien direct avec Jean-Michel Lagrange, tuteur du stage et responsable du développement des services informatiques et d'informations, j'ai établi un cahier des charges qui permet de lister les contraintes et besoins nécessaires à la vision d'ensemble du projet.

Sanios est une application web qui permet la gestion d'un parc automobile relatif aux véhicules de soin ou du milieu médical. Il permet aux personnels qualifiés de signifier des maintenances ou des hygiènes sur un véhicule donné. Il doit également pouvoir aux personnels désignés une vue d'ensemble sur les véhicules. Enfin, il doit offrir une vue dynamique et simple à l'ARS afin de s'assurer de l'entretien d'un véhicule. Il doit également permettre de générer des QR code et de produire des PDF.

Le projet s'articule autour de plusieurs profils définis offrant chacun un accès personnalisé à l'utilisateur. Certains existaient dans la version originale (les ambulanciers et l'ARS) d'autres répondaient à de nouveaux besoins de l'entreprise (Régulation, Gestionnaire et Mécanicien).

Conditions du stage

Après avoir examiné ma capacité à exécuter une veille technologie, mon adaptabilité et ma résilience face à l'autonomie, mon tuteur Mr Lagrange m'a proposé de travailler en distanciel partiel. J'ai donc, durant huit semaines sur dix de stage, été présente physiquement dans l'entreprise à Royan les Lundis et Jeudi uniquement.

Bien qu'ayant été autonome sur le projet, il était nécessaire que mon tuteur puisse consulter mon travail à tout moment afin de s'assurer de son avancement. C'est dans cette optique que nous avons partagé un projet Github, mis à jour deux fois par jours.

GitHub 

La plateforme de Microsoft Github a permis aux personnes concernées par le développement de Sanios de consulter son avancement et de faire des retours constructifs sur la direction du projet. Rapidement, j'ai utilisé le logiciel Github Desktop qui a simplifié mon utilisation de la plateforme. Motivé par le besoin de retour fréquents et la facilité d'utilisation du logiciel, c'est mon tuteur qui m'a dirigé vers cette interface.

Cahier des charges

Le cahier des charges du projet a été effectué du 25 Avril 2022 au 6 Mai 2022 (soit les deux premières semaines de stage).

Du fait de la spécialisation du secteur d'activité de l'entreprise, les termes employés dans le dossier ci-après correspondent aux termes usuels de l'entreprise Etoile Secours. Il était nécessaire dès la mise en place du cahier des charges du projet de définir un lexique.

L e x i q u e	
V.S.L :	Véhicule Sanitaire Léger.
Ambulance :	Ambulances de classe C
ASSU :	Ambulances contenant le matériel nécessaire aux soins d'urgences.
Roulants / Ambulanciers :	Conducteurs des véhicules incluant les V.S.L, les Ambulances et les ambulances de type ASSU.
ARS :	Autorité Régionale de la Santé.
Régulation :	Ambulancier dont le travail principal est de répondre au téléphone et d'organiser les déplacements de ses collègues Roulants.

Mécanicien :	Effectue des entretiens sur les véhicules de tous types.
Gestionnaire :	Administrateur du parc automobile.

Contraintes Globales

Le projet s'articule autour de multiples profils. Il était donc logique que les contraintes associées soient elles aussi réparties par profils. Pour représenter plus simplement et clairement ces exigences nous avons fait un tableau hybride inspiré des user stories.

En tant que (profil)
ROULANT / AMBULANCIER

Le Roulant scanne le QR CODE qui lui est relatif. Il arrive sur une interface :

- Il doit renseigner son nom, son prénom puis sa date de naissance
 - contraintes : le nom sera consulté dans une table nommée sanios_utilisateur de la base de données sanios. Les espaces contenus dans les noms seront transformés en '-'. Les accents contenus seront transformés en leur équivalence sans accents (ainsi "é" devient "e"). Les nom seront transformés en UPPERCASE
 - contraintes : le prénom sera consulté dans une table nommée sanios_utilisateur de la bdd sanios. Les espaces contenus dans les noms seront transformés en '-'. Les accents contenus seront transformés en leur équivalent sans accents (ainsi "é" devient "e"). Les prénoms seront transformés en LOWERCASE.
 - Si et seulement si le nom est reconnu dans la base de donnée, il est redirigé vers un nouveau formulaire. Il doit alors rentrer sa date de naissance. Si elle est erronée un message apparait.
 - contrainte : si un l'utilisateur 'ambulancier/roulant' tente une connexion sans scan de qr code, il est bloqué au niveau de la date de naissance.
- Il doit renseigner le type d'hygiène qu'il a effectué. (Nettoyage ou Désinfection)
 - Il ne peut rentrer qu'une seule hygiène à la fois.
 - contrainte : dans la table nommée sanios_hygiene de la database sanios, doit figurer l'heure et la date , l'id du véhicule, ainsi que le nom de l'utilisateur qui viens d'indiquer une hygiène sur un véhicule.
- Il doit pouvoir faire une liste des éléments présents dans le véhicule (masque etc...)
 - L'heure, la date , le nom de l'utilisateur et l'id du véhicules ainsi que son type doivent être ajoutés dans sanios_inventaire_(typevehicule)

En tant que (profil)
ARS

L'ARS scanne le QR CODE qui lui est relatif. Il arrive sur une interface :

- ▶ Peut consulter toutes les opérations d'hygiène sur 5 semaines. (du véhicule)
- ▶ Peut consulter ces données en PDF. Il peut aussi télécharger ce PDF.

En tant que (profil)

RÉGULE

- ▶ Accès à tous les véhicules de sa société.
- ▶ Accès pour un véhicule à ses données d'hygiène et d'entretien.
- ▶ Accès pour un véhicule à la date de son prochain controle technique
- ▶ Peut voir si un véhicule n'est pas localisé dans sa société. Si c'est le cas, il peut voir ou est le véhicule.
- ▶ Peut créer un véhicule (create sur sanios_vehicule)
 - A la création, il doit remplir les champs : plaque, ancien_id (qui lui est présenté sous le nom "alias"). Il doit choisir la marque et le modèle du véhicule.
 - contrainte : les modèles proposés doivent appartenir à une marque.
 - Il doit pouvoir ajouter une date de mise en circulation et une date de contrôle technique. Il doit également
- ▶ Va recevoir des alerte :
 - Alerte limite kilométrage avant cession du véhicule.
 - 300 000 Km pour véhicules BM, Renault, Skoda, etc ...
 - 500 000 Km pour véhicules Toyota
 - Alerte limite kilométrage avant vidange.
 - 10 000 Km
 - Alerte limite temps et/ou kilométrage avant distribution.
 - 100 000 Km
 - 5 ans
 -

En tant que (profil)

MÉCANICIEN

Il scanne le QR CODE relatif aux ambulanciers et arrive sur l'interface. Un bouton consacré lui propose de s'identifier en tant que mécanicien. Cette action l'amène à une interface réservée pour son profil.

- ▶ Peut renseigner le type d'entretien effectué.
- ▶ Récupération de la date et de l'heure lors de la validation de saisie d'un entretien.
- ▶ Il peut mettre à jour des véhicules (nombre de kilomètres uniquement)

En tant que (profil)

GESTIONNAIRE DE PARC AUTOMOBILE

- Peut consulter la position d'un véhicule.
- Peut consulter la liste de tous les véhicules par société.
- Peut consulter la liste des véhicules présents dans une société.

La différence entre la liste des véhicules par société et celle des véhicules présents dans la société réside dans le fait qu'un véhicule peut être dans une société à laquelle il n'est pas affilié. Permet de vérifier la position des véhicules.

- Peut consulter combien d'hygiène ont été faites par une personne.
- Accès pour un véhicule à ses données d'hygiène et d'entretien.
- Peut consulter la position de ses véhicules. (consulte les mêmes choses que le profil régule)
- Peut créer un véhicule (créer les mêmes choses que le profil régule)
- Il peut consulter l'historique de la mécanique véhicule. (accès au même tri que le profil mécanicien)
- Peut consulter toutes les interventions pour un véhicule (entretien et/ou hygiène)
- Va recevoir des alertes :
 - Alerte limite kilométrage avant cession du véhicule.
 - 300 000 Km pour véhicules BM, Renault, Skoda, etc ...
 - 500 000 Km pour véhicules Toyota
 - Alerte limite kilométrage avant vidange.
 - 10 000 Km
 - Alerte limite temps et/ou kilométrage avant distribution.
 - 100 000 Km
 - 5 ans
 -
- Peut modifier la cession d'un véhicule.
- Peut changer l'affectation d'une société.
- Alerte contrôle technique : si date contrôle technique existe alors date+2Y-15Jours
si date contrôle technique existe pas alors date+4Y-15J
- Peut sortir des statistiques sur le travail des mécaniciens (accès à sanios_entretien)

Spécificités Techniques

Voici la liste des spécifications techniques utilisées pour Sanios.

- **Visual Studio Code.**
- **phpmyAdmin**
- **Github Desktop**
- **PHP QR Code** est une librairie LGPL pour générer du QR Code
- export pdf : **FPDF**
- **Bootstrap** framework

Afin d'assurer la bonne mise en marche de Sanios, il était impératif de passer le projet sur le serveur de la société. La problématique s'est posée dès lors qu'il devenait nécessaire de vérifier le bon fonctionnement des qr codes.

De plus, il devenait nécessaire de constituer une base de données qui totalisait les 26 existantes en une unique.

Pour harmoniser notre travail, nous avons décidé avec Jean-Michel Lagrange d'utiliser php 7.4, wamp 3.2.6 et mysql 5.7. En effet, c'était la configuration requise par les serveurs utilisés au sein de l'entreprise.

Maquettage

S'il était nécessaire de repenser l'identité visuelle de Sanios, il était également important de considérer que son utilisation était devenue intuitive pour ses utilisateurs. Sur les profils d'Ambulanciers et d'ARS chaque modification majeure par rapport à l'ancienne version du site devait impérativement être validée par mon interlocuteur Mr. Lagrange.

Afin de simplifier et d'uniformiser la vue, nous avons décidé lors de l'établissement du cahier des charges d'utiliser le framework Bootstrap développé par Twitter. En effet, il permettait une implémentation rapide et précise grâce à son système de grille tout en laissant une certaine liberté dans son maniement.

Pour rester cohérent avec le milieu spécifique du soin et des ambulances, un code couleur a été mis en place. Il était relatif aux couleurs proposées par bootstrap mais a également été agrémenté de nouvelles couleurs décrites ci-dessous

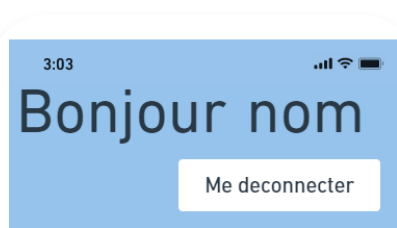
Pour réaliser les maquettes de l'application, j'ai utilisé le site whimsical (<https://whimsical.com/>). J'ai choisi ce créateur de wireframe pour plusieurs raisons :

- Avec une simple inscription, il permet de posséder un espace de travail sur lequel il est possible de réaliser divers projets.
- Il permet d'inviter un utilisateur à visualiser notre maquette
- J'ai eu l'opportunité de travailler avec auparavant et connaissait de ce fait l'interface ce qui simplifiait son utilisation.

Il m'a cependant également restreint sur les couleurs utilisées. En accord avec mon responsable la colorimétrie des maquettes a servi d'indicateur basique mais non relatif à la réalité du produit fini contraint lui par une charte graphique.

En considérant les besoins par profil, j'ai pu créer des wireframes qui correspondaient aux différents formats attendus. Je vous propose ci-après deux exemples de ces réalisations :

Les ambulanciers consultent SANIOS uniquement sur un format téléphone :



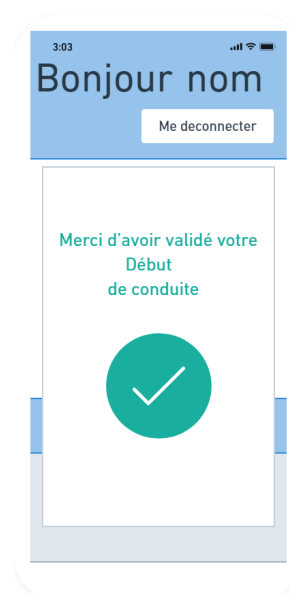
◀ Le header affiche un message de bienvenue en notifiant le nom de l'utilisateur. Il possède également un bouton pour se déconnecter.

◀ Le corps de l'interface propose deux boutons qui renvoient chacun sur une page dédiée ainsi que leurs labels respectifs.

◀ Le corps propose également une partie composée de deux autres boutons qui déclenchent une modale lors d'une action de l'utilisateur.

Le header reste le même lors de l'événement "click" sur le bouton "Début▶

Le corps du visuel est caché durant 3 secondes par une modale qui indique qu'un événement s'est déroulé tel qu'attendu. Au bout du temps imparti, elle disparaît.▶



◀ Le header de la page Hygiènes garde son message de bienvenue mais s'adapte pour proposer un bouton de retour signifié par une flèche

◀ Une nouvelle partie informe l'utilisateur sur l'identité du véhicule qu'il s'apprête à modifier.

Type d'Hygiène

Type 1
Type 2
Type 3
Type 4
Je valide

◀ Une série de bouton lui propose de choisir un type d'hygiène effectué

◀ Une fois choisie, il peut valider son entretien en cliquant sur “je valide”. Il sera alors redirigé vers l'accueil de son profil.

Le bouton choisi par l'utilisateur change d'apparence pour signifier sa sélection. ▶

Type d'Hygiène

Type 1

Type 2

Type 3

Type 4

Je valide

La régulation consulte SANIOS uniquement sur format ordinateur :

Bonjour nomSociete

Tous les véhicules modifier / créer un véhicule

Vehicules

tableau de bord marques chercher par alias

AA1 OPEL FK352FR

HYGIENES

Disinfection Appareils N/A

Disinfection Courante N/A

Disinfection Appareils 2022-09-01

Nettoyage Extérieur N/A

ENTRETIENS

vidange date personnel

parebrise date personnel

carrosserie date personnel

type entretien date personnel

AA2 OPEL FK352FR

AA2 OPEL FK352FR

Vehicule location

Date de controle technique

Bonjour nomSociete

Tous les véhicules modifier / créer un véhicule qr code

IMMATRICULATION AMBULANCIERS

IMMATRICULATION ARS

Télécharger le pdf

Bonjour nomSociete

Tous les véhicules modifier / créer un véhicule

Modifier un véhicule

nom - immatriculation - marque - modele

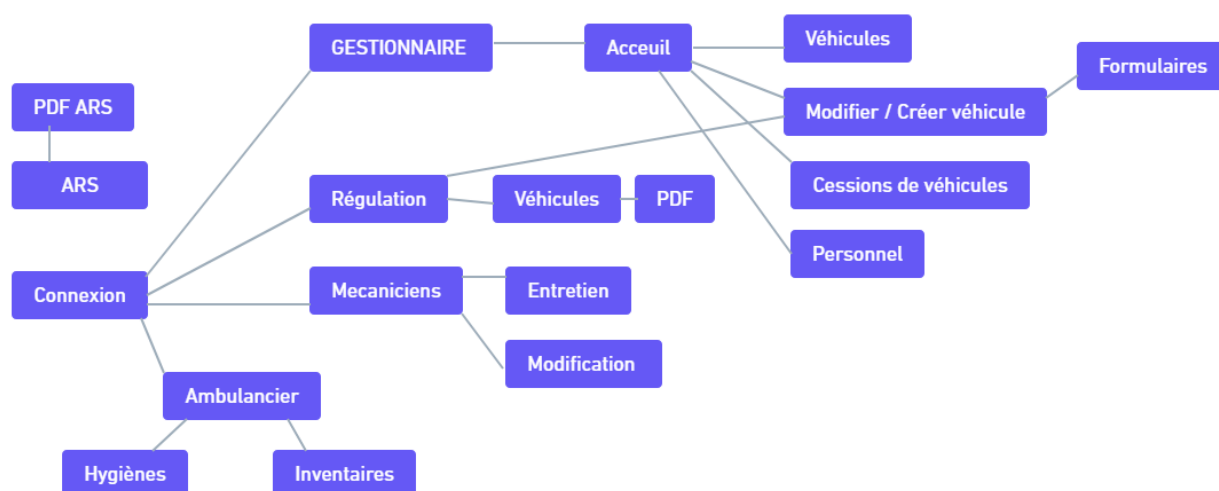
modifier

Créer un véhicule

marque

créer

Les maquettes effectuées n'ont cependant pas toutes été relatives aux vues utilisateurs. Le projet était suffisamment conséquent pour constituer un mapping de l'expérience utilisateur. Ce mapping permettait de définir l'architecture du site Sanios. Ainsi, cette cartographie de l'application web a permis de dégager un schéma à utiliser pour l'implémentation (Model Vue VueModel) de l'application web mais elle a également été utile à rentabiliser le temps de développement.



Responsif

Le projet Sanios est destiné à être vu par des formats prédéfinis. Ainsi, en fonction du profil de l'utilisateur, un seul et unique format sera appliqué. Sont consulté par un format portable / téléphone les profils mécaniciens, ambulanciers, ars. Sont consulté par un format ordinateur les profils régulation et gestionnaire.

Nous pourrions noter que le format tablette reste absent des formats pris en compte, c'est une volonté du maître de stage que d'ignorer simplement ce format qui ne correspond à aucun besoin de l'entreprise.

Pour illustrer l'adaptation au format, je vous propose de nous focaliser sur une page qui fait appel à une feuille de css spécialement dédiée au changement de format.

Voici la page index.php de Sanios en format ordinateur (laptop 1024x882)

VOTRE NOM

VOTRE PRENOM

Implémenté via le framework bootstrap grâce aux liens :

```

<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">

```

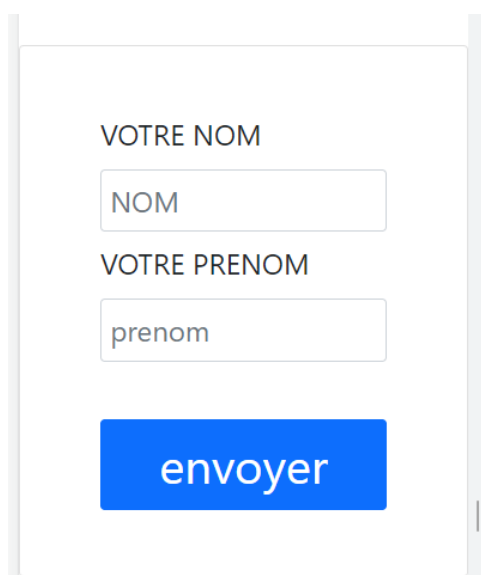
la balise `link` permet de définir la relation entre le document courant et une ressource externe (ici bootstrap). Grâce à cet ajout de feuilles de style, il est désormais possible d'utiliser des classes propres au framework.

Notre header se compose ainsi :

```
<div class="container my-5">
  <div class="row ">
    <div class="card">
      <div class="card-body">
        <!-- Ici notre contenu : le formulaire. -->
      </div>
    </div>
  </div>
</div>
```

`container` possède des valeurs de `margin` sur l'axe `x` qui lui permettent de se centrer.
`my-5` permet de définir un `margin` de `3rem` sur l'axe `y` (le bas et le haut du conteneur).
`card` permet de définir une `width` de `100%` (prend toute la largeur de son élément parent).

Nous pouvons constater que l'élément central prend ainsi toute la largeur de la page en gardant un espacement sur les côtés. Regardons à présent la vue portable :



Sous le format portable (375x882), les marges sur les cotés disparaissent, le texte est grossi mais reste sur une unique ligne. Pour effectuer ses changements, le code dispensé par le framework a dû être modifié. En ajoutant :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="vue/media/css/breakpointIndex.css">
```

La vue portable est implémentée. En plaçant la balise de métadonnées <meta> avec l'attribut viewport, nous donnons des informations relatives à la taille et l'échelle du viewport aux navigateurs mobiles afin que les différents éléments d'une page s'affichent mieux. Le contenu de la page breakpointIndex.css sert à modifier certaines classes bootstrap pour adapter leurs contenu :

```
@media only screen and (max-width: 600px) {
  .card-body {
    padding: 1% !important;
  }

  .card-title{
    font-size: 1.5rem !important;
  }
  .card-text{
    font-size: 1.5rem !important;
  }

  .mb-2{
    margin-bottom: .5rem !important;
  }
  input{
    margin-top: 5% !important;
    padding: 3% !important;
    width: 100% !important;
    font-size: 2.5rem !important;
  }

  .mt-3{
    margin-top: 3rem !important;
  }
}
```

En ajoutant la mention “!important” le navigateur va en priorité appliquer cette redéfinition du style.

la ligne “@media only screen and (max-width: 600px)” permet de contraindre ses changements uniquement aux formats en dessous du largeur de 600px.

Voici la modification d'une ligne (ici la redéfinition de la classe card-text en ajoutant la ligne font-size : 1.5rem !important)

Vue sans la ligne	Vue avec la ligne
<div>VOTRE NOM</div> <div>NOM</div> <div>VOTRE PRENOM</div> <div>prenom</div>	<div>VOTRE NOM</div> <div>NOM</div> <div>VOTRE PRENOM</div> <div>prenom</div>

Cette modification permet d'adapter le texte au format rendant ainsi la lecture plus simple et claire.

*Merci de consulter l'annexe intitulée Cahier des Charges pour voir l'ensemble du cahier des charges.

Sécurité et protection des formulaires.

Sanios étant une application web interne à l'entreprise Etoile Secours, il était décisif de sécuriser les formulaires et plus particulièrement les formulaires de connexion. Si la vue de l'ARS reste ouverte à tout utilisateur scannant le QR Code dédié, les autres vues doivent dépendre de certaines conditions qui sont autant de sécurités.

Une sécurité par harmonisation des données : Puisque dès la première page de l'application web l'utilisateur est invité à entrer des données et qu'à la validation de ce premier formulaire une requête SQL est envoyée sur la base de données, il était obligatoire de protéger ces champs.

```
if (isset($_POST["send"])) {  
    if (!empty($_POST["nom"]) && !empty($_POST["prenom"])) {  
        require __DIR__.'../../model/indexModel.php';  
    }  
}
```

```

        $name = cleanString($_POST['nom'], $format[0]);
        $lastname = cleanString($_POST['prenom'], $format[1]);
        //Appel de BDD
        $userData = IndexFetch($name,$lastname);

        if($userData) {
            //Lancement de la session
            $_SESSION['logged'] = true;
            //Données de Session
            $_SESSION['id_user'] = $userData['id_user'];
            $_SESSION['nom'] = $userData['nom'];
            $_SESSION['prenom'] = $userData['prenom'];
            $_SESSION['mdp'] = $userData['mdp'];
            $_SESSION['date_naissance'] = $userData['date_naissance'];
            $_SESSION['fk_societe'] = $userData['fk_societe'];
            $_SESSION['fk_emploi'] = $userData['fk_emploi'];
            $_SESSION['statut'] = $userData['statut'];
            $profil = $_SESSION['fk_emploi'];
        }

    } else {
        $_SESSION = array();
        session_destroy();
    }
}

```

Je signifie en premier lieux que le bouton avec l'attribut name="send" as bien été activé, puis je vérifie que les champs sont bien remplis. Si cette seconde condition est négative, je vide la session (créée à l'arrivée sur la page) et la détruit. Le \$_POST[""] de php rafraîchit la page ce qui permet d'ouvrir une nouvelle session pour essayer de se connecter de nouveau.

Si mes conditions sont remplis alors je passe le résultat de mes input dans une fonction nommée cleanString qui prend en paramètre le contenu du champs input ainsi qu'une cellule de tableau nommé format[] (\$format[0] = 'MAJUSCULE', \$format[1] = 'MINUSCULE'). Une fois ce traitement fait, la fonction IndexFetch() va pouvoir aller demander à la base de donnée qui est l'utilisateur et remplir les valeurs dans les variables de session.

La fonction cleanString est la suivante :

```

/**
 * Function cleanString
 * @param string $str

```

```

* @param string $format
* @return string $str
*/
function cleanString($str, $format)
{
    $str = htmlspecialchars($str);
    $str =
    html_entity_decode(preg_replace('/&([a-zA-Z]{1,2})(?:acute|cedil|circ|grave|l
ig|orn|ring|slash|th|tilde|uml);/i', '$1', htmlentities($str, ENT_QUOTES,
'UTF-8')), ENT_QUOTES, 'UTF-8');
    $str = trim(preg_replace('/^[^A-Za-z]+/i', '-', $str), '-');
    if ($format == "MAJUSCULE") {
        $str = mb_strtoupper($str);
    } else {
        $str = mb_strtolower($str);
    }
    return $str;
}

```

Afin de se prémunir des failles XSS (exécution de scripts cotés client), cette fonction utilise la fonction php htmlspecialchars. Elle permet de convertir des caractères usuellement utilisés dans des scripts par des caractères de remplacement.

Pour continuer dans cette logique de protection, la fonction php html_entity_decode permet de décoder les entités. avec en paramètre la fonction php preg_replace cela permet de contraindre des lettres uniquement puis remplace les accents et autres espaces en lettre. La fonction trim avec preg_replace permet quand à elle de changer les espaces en tirets.

Ainsi “Jean Benoît” entré dans le champ input nom ressortira dans la base de donnée en temps que “JEAN-BENOIT”.

Une sécurité par identification : Afin de pouvoir se connecter sur Sanios, il est indispensable d’être dans la base de données en temps qu’utilisateur. Si un utilisateur n’est pas présent dans la base de données, il est alors ignoré et reste sur la page de connexion jusqu’à ce que les données soient valides.

Une sécurité par mots de passe : Si l’accès au profil ARS diffère des autres, les profils possèdent tous un mot de passe. Qu’il s’agisse d’un mot de passe crypté en SHA256 dans la base de donnée ou d’une date d’anniversaire (contraintes imposés par l’entreprise) . Se tromper de mot de passe ou rentrer une valeur non référencée dans la base de donnée entraine l’apparition d’un message d’erreur.

Votre mot de passe est incorrect

Une sécurité par type de profil : Les mots de passe sont également vérifiés au niveau de la base de donnée en regardant le profil de l'utilisateur, ainsi ce n'est pas parce qu'un mot de passe est présent dans la base de données qu'il peut être utilisé pour n'importe quel profil. Le profil est intimement lié au nom, au prénom et au mot de passe de l'utilisateur.

Une sécurité par présence de données : Certains profils tels que le mécanicien ou l'ambulancier doivent, pour s'enregistrer scanner un QR Code. Ce dernier envoie la plaque d'immatriculation au navigateur via un \$_GET il est possible de la récupérer et de la stocker. Ces utilisateurs précis n'ont pas de raison d'aller sur leur interface si aucun véhicule n'a été scanné aussi en l'absence de cette variable l'accès à leur profil est refusé.

Développer une interface utilisateur web dynamique

L'implémentation de bootstrap permet également l'ajout d'éléments de dynamisme. L'utilisation de menu dropdown est utilisée par exemple dans la vue du profil gestionnaire ou régule. Pour implémenter ces éléments, il était indispensable d'ajouter juste avant la fermeture de la balise <body> les lignes suivantes :

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

La première permet d'ajouter le javascript de bootstrap, la seconde le jquery.

Pour garder la main sur certains composants dynamique, j'ai également créé une modale pour le profil "roulant" Je vous propose de décortiquer son fonctionnement.

```
1. <script>
2. /** Création d'une modale avec des événements au click. Ajout d'un 3.
   texte spécifique à l'action effectuée dans la modale. */
4. var modale = document.getElementById("ModalValidate");
5. var StartToggle = document.getElementById('btnDebut');
6. var EndToggle = document.getElementById('btnFin');
7. var innerText = document.getElementById("innerText");
8.
9. StartToggle.addEventListener('click', function(){
10.   modale.classList.toggle("hidden");
```

```

11. var textBox = " début ";
12. innerText.innerHTML = textBox;
13.
14. setTimeout(() => {modale.classList.toggle("hidden"); }, 3000);
15. });
16.
17. EndToggle.addEventListener('click', function(){
18.   modale.classList.toggle("hidden");
19.   var textBox= " fin ";
20.   innerText.innerHTML = textBox;
21.   setTimeout(() => {modale.classList.toggle("hidden"); }, 3000);
22. });
23.
24. </script>

```

Voici un exemple qui utilise ces effets de dynamisme.

I.4 : Une nouvelle variable nommée modale est créée qui récupère un élément du HTML qui porte l'attribut id="ModalValidate".

I.5 : Une nouvelle variable nommée StartToggle est créée qui récupère un élément du HTML qui porte l'attribut id="btnDebut".

I.6 : Une nouvelle variable nommée EndToggle est créée qui récupère un élément du HTML qui porte l'attribut id="btnFin".

I.7 : Une nouvelle variable nommée innerText est créée qui récupère un élément du HTML qui porte l'attribut id="InnerText".

I.9 Ajout d'un écouteur d'événement sur la variable StartToggle. Ici l'événement est un click utilisateur. Cet événement déclenche une fonction.

I.10 La fonction déclenchée par l'événement provoque un changement de classe sur la variable modale. Si l'élément ciblé par cette variable possède la classe "hidden" il la retire, si elle ne la possède pas, il l'ajoute.

I.11 création d'une variable nommée textBox qui contient le string " début ".

I.12 Insertion du contenu de la variable textBox à l'intérieur de la variable innerText.

I.14 Définition d'une fonction timer. Le comportement est en attente durant 3 secondes. Au bout du temps imparti, la variable modale change de nouveau de classe.

Ainsi au click sur le bouton début le résultat attendu est le suivant :

Merci d'avoir validé votre
Début
de conduite



◀ Au click sur le bouton la fonction retire la classe hidden (qui contient uniquement un display none) de la modale affichant le bloc au milieu de la page.

◀ Le texte correspondant est affiché afin de renseigner quel comportement vient d'avoir lieu. Ici l'utilisateur a cliqué sur "Début"

◀ Au bout de trois secondes, le timer ajoute la classe hidden pour cacher de nouveau l'élément.

Le second évènement qui agit sur la variable EndToggle (l.17 à 20) possède le même comportement que sa consœur, la différence notable viens de la variable textBox qui prend cette fois ci la valeur " fin ". Elle est également déclenché par un click sur un bouton différent.

Cette duplicité de la fonction permet facilement d'afficher le message correcte en fonction du bouton sélectionné.

Définir une base de données cohérente

Le constat du responsable de l'informatique de la société était clair : trop de bases de données éclatées entraînait une impossibilité de croiser certaines données. Il rendait également difficile de réaliser un suivi efficace des véhicules. Par exemple, nous avons pu remarquer que certains véhicules disparaissaient purement et simplement des bases de données puisque ces dernières étaient rattachées à une unique société tandis que les véhicules eux pouvaient se déplacer de sociétés en sociétés. Cette inadéquation rendait difficile un suivi efficace de la localisation des véhicules.

Le but de ce nouveau Sanios était en priorité d'harmoniser les données afin de pouvoir les consulter simplement. La création de la base de données a donc été primordiale. De plus, il était nécessaire qu'elle s'adapte aux nouvelles fonctionnalités du service.

Nous avons agit en plusieurs étapes. En premier nous avons identifiés les tables que nous pouvions regrouper. Nous avons pu ainsi créer sur une base de donnée nommée sanios ajouter les tables suivantes :

sanios_véhicules(id_vehicule,plaque,#fk_marque,#fk_modele,#fk_type,#fk_societe,
#fk_inventaire_position,date_mise_circulation,date_controle_technique,kilometrage,cession,ancien_id)
sanios_marque (id_marque, marque_nom)
sanios_modele (id_modele ,fk_marque, model_nom)
sanios_type (id_type, type_nom)

sanios_societe (id_societe, societe_nom)

Chacunes des tables (societe, marque, modele, type) comportent la même structure. Ce n'est pas un hasard. Il était plus évident de les isoler par des identifiants et de leurs donner les mêmes champs afin de pouvoir dans les requêtes php n'utiliser qu'une unique fonction qui permettait par la suite de toucher n'importe laquelle de ces tables. Par exemple la fonction nameFetch tirée de la page gestionnaireModel.php :

```
/**
 * nameFetch function
 * Retourne une requette qui récupère tous les noms de marque / modele / type
 * @param [string] $param_nom : concaténation pour reconstituer la requette
 * @param [string] $param : "marque" OU "modele" OU "type"
 * @return iterable
 */
function nameFetch($param) {
    $bd = connectBd();
    $param_nom = $param . "_nom";
    $getParamSql = $bd->prepare("SELECT $param_nom from sanios_$param ");
    $getParamSql->execute();
    $getParam = $getParamSql->fetchAll(PDO::FETCH_ASSOC);
    return $getParam;
}
```

Cette fonction s'appelle avec un string "marque" ou "modele" ou "type". Grâce à une concaténation, elle permet de récupérer pour chaque table les nom associés.

Ainsi avec l'appel :

```
$marques = nameFetch('marque');
```

La variable marques contient tous les noms des marques inscrites dans la base de données.

Ci dessous, la composition complète de la base de données.

sanios_utilisateur (id_user, nom, prenom, mdp, #fk_societe,#fk_emploi, date_naissance)

sanios_emploi (id_emploi, emploi_nom)

sanios_hygiene (id_entretien,#fk_vehicule,#fk_user,date,#fk_user_societe,#fk_vehicule_societe#fk_type)

sanios_hygiene_type (id_hygiene_type, nom)

sanios_entretien (id_mecanique,#fk_vehicule,#fk_user,date,heure,#fk_user_societe,#fk_vehicule_societe,#fk_kms,#fk_type,autre,externe)

sanios_entretien_type (id_entretien_type, nom, alerte)
sanios_inventaire_ambu(id_inventaire,date_inventaire,#fk_id_vehicule,#fk_id_user,#fk_id_user2,10P, 10D ,20P,20D, 30P,30D, [...] , 460P,460D, 470P,480D)
sanios_inventaire_assu (id_inventaire,date_inventaire,#fk_id_vehicule,#fk_id_user,#fk_id_user2,10P, 10D ,20P,20D, 30P,30D, [...] , 530P, 530D, 540P,550P)
sanios_inventaire_vsl (id_inventaire,date_inventaire,#fk_id_vehicule,#fk_id_user,#fk_id_user2,10P, 10D ,20P,20D, 30P,30D, [...] , 310P,310D, 320P, 330P)
sanios_liste_inventaire_ambu (id_produit,nom,quantite)
sanios_liste_inventaire_assu (id_produit,nom,quantite)
sanios_liste_inventaire_vsl (id_produit,nom,quantite)
sanios_marque (id_marque,marque_nom)
sanios_modele (id_modele,fk_marque, modele_nom)
sanios_societe (id_societe, societe_nom)
sanios_type (id_type, type_nom)
sanios_conduite (id, fk_user, fk_vehicule,date,conduite)
sanios_vehicule(id_vehicule,plaque,#fk_marque,#fk_modele,#fk_type,#fk_societe, #fk_inventaire_position,date_mise_circulation,date_controle_technique,kilometrage,cession,ancien_id)

*processus de création de la base de donnée dans l'annexe cahier des charges

Développer les composants d'accès aux données

Le site Sanios étant hébergé par les serveurs de l'entreprise, la mise en place d'un point de connexion était indispensable. Présent dans le projet sous le nom connexion.php, il définit les variables utilisées pour se connecter et implémente une fonction nommée connectBd() appelée dans tout le projet pour accéder à la base de données.

Dans l'exemple ci-dessous les noms des variables ont été masqués afin de ne pas divulguer des informations.

```

<?php
//Définition des variables
define ("SERVEUR", "*****");
define ("USER", "*****");
define ("MDP", "*****");
define ("BD", "sanios");

function connectBd() {
    try {
        $db = new PDO('mysql:host='.SERVEUR.';dbname='.BD,USER,MDP);
        $db->exec("SET CHARACTER SET utf8");
    }
  
```

```

catch (Exception $e)
{
    $db = 'Erreur : ' . $e->getMessage() . '<br />';
    $db = 'N° : ' . $e->getCode() ;
}
return $db;
}

```

Ici, grâce à l'objet PDO de php la fonction établit une connexion avec le serveur et la base de données. Puis la variable qui récupère ce point de connexion passe dans une nouvelle fonction permettant de supporter l'UTF8

Enfin, le try{}catch{} permet de visualiser ou signifier les erreurs éventuelles que pourrait rencontrer cette tentative de connexion. La fonction retourne donc soit un message d'erreur, soit un point de connexion.

Étant en modèle MVVM, réaliser des CRUD (Create, Read, Update, Delete) s'est rapidement montré intuitif. En effet, les fichiers vueModel sont des CRUD. Leur fonction unique est de faire le lien entre le point de connexion et le traitement des données. Ainsi, leur contenu est presque uniquement une succession de fonctions effectuant des requêtes SQL.

Pour illustrer cette réalité, je vous propose de regarder plus en détails les fichiers, ars.php (dans la vue) , arsVueModel.php (dans les vueModel) et arsModel (dans les model) pour comprendre le fonctionnement de ces CRUD.

La vue ARS définit une variable :

```
$immatriculation = $_GET['alias'];
```

Grâce à un require, cette vue appelle le fichier arsVueModel.php :

Ce nouveau fichier définit une nouvelle variable qui cette fois ci appelle une fonction avec en paramètre la variable précédente :

```
$vehicule = getDataVehicule($immatriculation);
```

De nouveau le fichier contient un require qui lui appelle le fichier arsModel.php

C'est dans ce fichier qu'est déclaré et définie la fonction

```

/**
 * getDataVehicule fonction
 * Récupère toutes les informations sur un véhicule grâce à sa plaque
 * @param [string] $immatriculation
 * @return itterable
 */
function getDataVehicule($immatriculation){

```

```

    $bd = connectBd();
    $sql = $bd->prepare("SELECT * from sanios_vehicule JOIN sanios_marque on
fk_marque = id_marque
    JOIN sanios_societe ON fk_societe = id_societe where plaque = :immat ");
    $sql->bindParam('immat',$immatriculation,PDO::PARAM_STR);
    $sql->execute();
    $retourSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retourSql;
}

```

Ce fichier quand à lui appelle le fichier qui contient le point de connexion et l'implémente via la fonction connectBd().


Interconnexion du code

Si les profils étaient différents les uns des autres et nécessitent chacun des traitements isolés, certaines fonctionnalités étaient utilisées par plusieurs profils. Il m'est rapidement apparu que le projet allait demander de dupliquer beaucoup de code. La structure MVVM du projet a permis de réduire cette duplication inutile et lourde. Grâce à la création de fonctions dédiées, il était possible de contraindre certaines utilisations en réalisant un traitement global dans le même temps.

L'un des meilleurs exemples pour illustrer ce fonctionnement est le formulaire de modification de véhicule. Utilisé par la régulation et par le Gestionnaire, il ne peut pas proposer les mêmes informations ou les mêmes choix aux deux profils mais doit effectuer la même tâche.

L'arrivée sur les vues de la page intitulée formulaire.php sont les suivante :

La vue du profil Régule :


[Me deconnecter](#)

[Voir les véhicules](#)
[Modifier / Créer un véhicule](#)

MODIFIER LE VEHICULE - TEST

Ambulances ASUR

Marque du véhicule

Modele du véhicule


Type du véhicule

Plaque d'immatriculation

Date de mise en circulation

Je valide mes modifications

La vue du profil Gestionnaire


[Me deconnecter](#)

[Voir tous les véhicules](#)
[Modifier / Créer un véhicule](#)
[Voir les véhicules cédés](#)
[Voir le personnel](#)

MODIFIER LE VEHICULE - TEST

Société du véhicule

Marque du véhicule

Modele du véhicule

Type du véhicule

Plaque d'immatriculation

Inventaire Position

Date de mise en circulation

Cession du véhicule

Je valide mes modifications

Si le bandeau de navigation appartient à une page extérieure appelée par un require, le corps de la page est le même mais est modifié en fonction du profil. Cette page demande un traitement supplémentaire aux autres puisqu'elle doit définir et sécuriser le profil actuel arrivé sur cette page.

```

<?php
(!isset($_SESSION)) ? session_start() : session_abort();
if($_SESSION['fk_emploi'] != 4 && $_SESSION['fk_emploi'] != 3 ) {
    header('location: ../index.php');
}
if($_SESSION['fk_emploi'] == 4){

```

```

        (!isset($_SESSION)) ? session_start() : session_abort();
        $accueil = 1;
        $colorBtn = "secondary";
        $returnProfil = "gestionnaireForm.php";
        require __DIR__.'./gestionnaire.php';
    }
    if ($_SESSION['fk_emploi'] == 3) {
        (!isset($_SESSION)) ? session_start() : session_abort();
        $accueil = 1;
        $colorBtn = "primary";
        $returnProfil = "reguleForm.php";
        require __DIR__.'./regule.php';
    }
    require ('../vuemodel/formulaireVueModel.php');
?>

```

Ainsi, la page va passer par plusieurs validations.

La première est un ternaire qui vérifie si l'utilisateur fait partie d'une session.

Si ce n'est pas le cas, la session est lancée, sinon, elle est momentanément interrompue pour éviter des soucis de multiple ouvertures de sessions.

Une fois cette vérification effectuée, nous vérifions si les profils de cette session (4 pour gestionnaire et 3 pour régulation) correspondent aux profils attendus. Il n'y a aucune raison pour qu'un utilisateur non loggé ou avec le mauvais profil puisse arriver sur cette page, si ces condition ne sont pas remplies, l'utilisateur est redirigé directement sur la page de connexion.

Puis, par profil, nous déterminons des variables qui serviront de constantes tout au long du formulaire (la couleur des boutons bootstrap ou l'adresse du lien de redirection.).

Ainsi tout au long de sa structure HTML, cette page "jongle" entre les profils pour afficher uniquement à l'utilisateur les informations qui lui sont dédiées. (Annexe démonstration du formulaire de création.)

Le session_abort()

Revenons sur le sessions_abort() utilisé dans la fonction ci-dessus décrite.

Dans notre exemple, était importé grâce à un require le bandeau lié au profil actuel (variable de session). Cependant, dans cette nouvelle page, il est nécessaire de consulter cette session pour pouvoir effectuer des vérifications de session à cet endroit précis pour

protéger le formulaire de tout accès impromptu. Rouvrir une session était impossible puisque ouverte précédemment mais détruire la session avec un `session_destroy()` n'était pas souhaitable non plus. J'ai donc consulté le site anglophone stack overflow pour trouver une alternative mais également pour vérifier que cette solution était optimale.

Voici les exemples que j'ai trouvés lors de ces recherches :

▲ `session_abort()` is analogous to [session_write_close\(\)](#).

9 PHP locks session data during a web request to prevent data corruption on multiple simultaneous requests.

▼

✓ When **Request 1** comes in, **Session 1** is locked by that process so it can make any changes needed. If **Request 2** comes in for **Session 1**, php blocks until the session lock is released to make sure that **Request 2** has the most up-to-date session data.

🕒

`session_abort()` closes the session and releases the lock without flushing session data to the session storage mechanism while `session_write_close()` writes the current content of session back and then closes / releases the lock.

edit: Calling `session_abort()` or `session_write_close()` will let php process **Request 2** even if **Request 1** is not finished processing.

Share Follow

edited Sep 14, 2015 at 15:12

answered Sep 14, 2015 at 13:40



Josh J

6,673 ● 3 ● 21 ● 44

Traduction :

`session_abort()` est analogue à [session_write_close\(\)](#).

PHP verrouille les données de session lors d'une requête Web pour empêcher la corruption des données lors de plusieurs requêtes simultanées.

Lorsque la **requête 1** arrive, la **session 1** est verrouillée par ce processus afin qu'elle puisse apporter les modifications nécessaires. Si la **requête 2** arrive pour la **session 1**, php se bloque jusqu'à ce que le verrou de session soit libéré pour s'assurer que la **requête 2** dispose des données de session les plus à jour.

`session_abort()` ferme la session et libère le verrou sans vider les données de session dans le mécanisme de stockage de session tandis que `session_write_close()` réécrit le contenu actuel de la session, puis ferme/libère le verrou.

edit :

modifier : appeler `session_abort()` ou `session_write_close()` permettra à php de traiter la requête 2 même si la requête 1 n'est pas terminée.

répondu par Josh J le 14 Septembre 2015

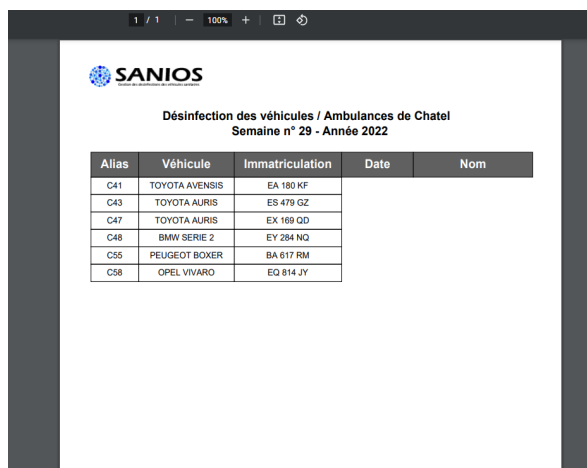
Source : [purpose-of-session-abort](#)

Si cette réponse à la question “quelle est l'utilité du `session_abort()`” est datée, elle reste, selon mes recherches, pertinente. Elle m'a permis de “locker” ou verrouiller la session le temps de faire des vérifications tout en gardant en mémoire ses données. Puis de la rouvrir lorsque j'en avais besoin sans pertes.

Génération de PDF

Lors de l'établissement du cahier des charges du projet Sanios, mes échanges avec mon tuteur Mr. Lagrange ont montré qu'il était prioritaire que l'ARS (Agence Régionale de la Santé) puisse avoir accès aux données sous forme d'un pdf téléchargeable. Aucune contrainte ne m'était imposée quant à la forme du document, j'ai cependant décidé de garder dans mon inspiration les fichiers générés par l'ancienne interface. Dans un souci de simplicité et de facilité de lecture, il était incohérent de changer complètement de visuel.

Ancien Sanios pdf ARS



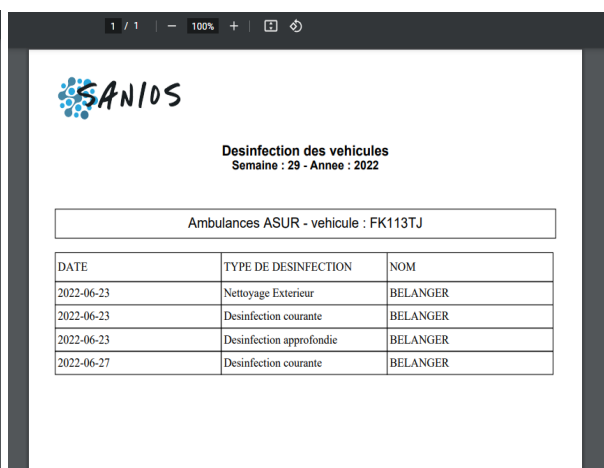
1 / 1 | 100% + | [Icon] [Icon]

SANIOS
Service de désinfection des véhicules

Désinfection des véhicules / Ambulances de Chatel
Semaine n° 29 - Année 2022

Alias	Véhicule	Immatriculation	Date	Nom
C41	TOYOTA AVENSIS	EA 180 KF		
C43	TOYOTA AURIS	ES 479 GZ		
C47	TOYOTA AURIS	EX 169 QD		
C48	BMW SERIE 2	EY 284 NQ		
C55	PEUGEOT BOXER	BA 617 RM		
C58	OPEL VIVARO	EQ 814 JY		

Nouveau Sanios pdf ARS



1 / 1 | 100% + | [Icon] [Icon]

SANIOS

Desinfection des vehicules
Semaine : 29 - Année : 2022

Ambulances ASUR - vehicule : FK113TJ

DATE	TYPE DE DESINFECTION	NOM
2022-06-23	Nettoyage Extérieur	BELANGER
2022-06-23	Desinfection courante	BELANGER
2022-06-23	Desinfection approfondie	BELANGER
2022-06-27	Desinfection courante	BELANGER

De plus les bases de données ayant été commutées en une unique, nous avons décidé de proposer un pdf accessible sur les profils régulation et gestionnaires qui affichaient les Qr

Code relatifs au véhicule sélectionné. Ainsi, il devient possible de télécharger de nouveaux Qr Code si un souci intervient sur le véhicule.

Pour générer ces PDF mon tuteur m'a proposé d'utiliser la librairie FPDF. Après avoir téléchargé ce nouveau composant et l'avoir inclus dans mon projet, j'ai mis en place la gestion des PDF.

L'appel au fichier de génération de PDF (pdfgenerator.php) se fait via un bouton sur le profil Gestionnaire ou Régulation. Ce bouton redirige vers la page :

pdfGenerator.php?roulantpath=../vue/media/qrCode/QrcodeIdRoulantFK113TJ.png&arspath=../vue/media/qrCode/QrcodeIdARSFK113TJ.png&id=FK113TJ

Il s'agit de la page, puis de variables passées en \$_GET (donc dans l'url). Ces dernières contiennent deux strings (roulantpath et arspath) ainsi qu'une variable contenant un id.

```

1.1 require('../fpdf184/fpdf.php');
1.2 $pathRoulant= $_GET['roulantpath'];
1.3 $pathARS = $_GET['arspath'];
1.4 $immatriculation = $_GET['id'];
1.5 class PDF extends FPDF
1.6 {
1.7 // En-tête
1.8 function Header()
1.9 {
1.10 // Police Arial gras 15
1.11 $this->SetFont('Arial','B',15);
1.12 // Décalage à droite
1.13 $this->Cell(70);
1.14 // Titre
1.15 $this->Cell(40,15,'SANIOS',1,0,'C');
1.16 // Saut de ligne
1.17 $this->Ln(20);
1.18 }
1.19
1.20 // Pied de page
1.21 function Footer()
1.22 {
1.23 // Positionnement à 1,5 cm du bas
1.24 $this->SetY(-15);
1.25 // Police Arial italique 8
1.26 $this->SetFont('Arial','I',8);
1.27 // Numéro de page

```

```

1.28     $this->Cell(0,10,'Page '.$this->PageNo().'{nb}',0,0,'C');
1.29 }
1.30
1.31 }
1.32
1.33 // Instanciation de la classe dérivée
1.34 $pdf = new PDF();
1.35 $pdf->AliasNbPages();
1.36 $pdf->AddPage();
1.37 $pdf->SetFont('Arial','',14);
1.38 $pdf->Image($pathRoulant,40,40);
1.39 $pdf->Image($pathARS,130,40);
1.40
1.41 $pdf->SetMargins(40,0,0);
1.42 $pdf->Ln(0);
1.43 $pdf->Cell(100,90,$immatriculation,0);
1.44 $pdf->Cell(10,90,$immatriculation,0);
1.45 $pdf->Ln(5); // saut de ligne 10mm
1.46 $pdf->Cell(100,90,'ROULANT',0);
1.47 $pdf->Cell(100,90,'ARS',0);
1.48 $pdf->Output();

```

La première étape de la création du pdf est d'inclure la librairie au fichier (l.1).

Puis nous récupérons les variables pour les stocker de manière pérenne dans de nouvelles variables (l.2 à l.4)

De l.5 à l.31, nous définissons une classe qui va créer des composants de notre PDF en se servant des définitions déclarées dans la librairie (grâce au mot clé `extend`). Cette fonction définit un header et un footer.

l.34 permet la création d'un nouvel objet appelé `$pdf`. Cette nouvelle instance va nous permettre d'accéder aux fonctionnalités de notre classe pdf mais également à celles de sa classe mère

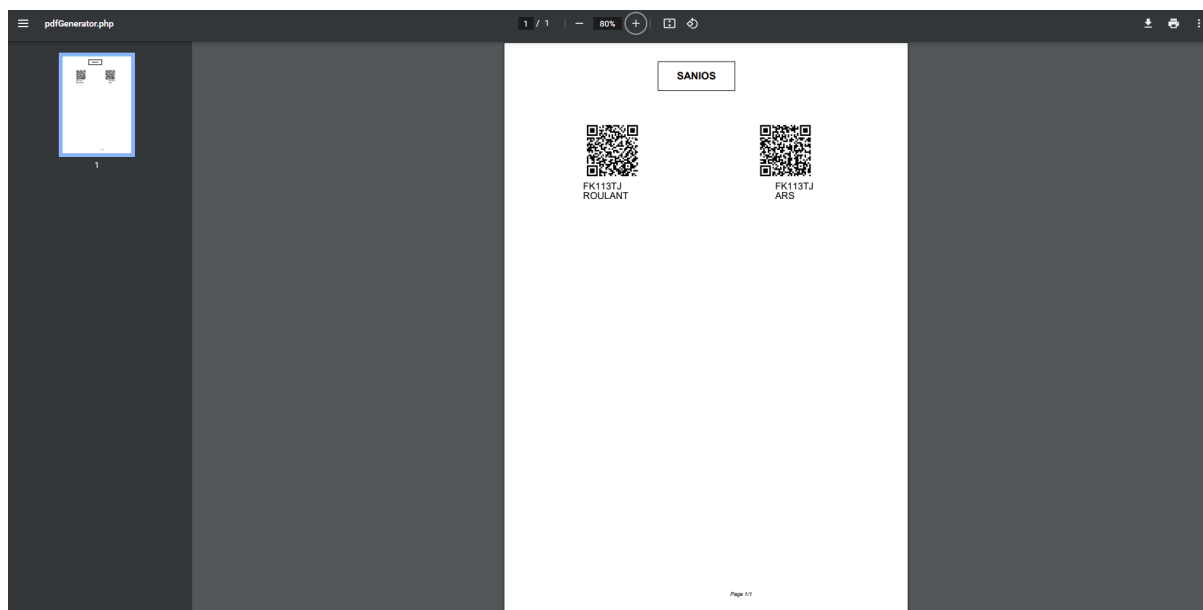
l.35 : nous lui attribuons un numéro de page. (Ici il n'y a aucune raison d'avoir dans notre PDF plus d'une page puisqu'il ne va contenir que deux qr code et du texte.

l.38 à 39 Nous ajoutons les images (png) de nos qr code dans le pdf à une place destinée.

l.41 à l.48 : Il s'agit de faire de la mise en forme de notre document, nous ajoutons des marges puis du texte sous les images des qr codes.

Enfin en l.48 nous exécutons la fonction `Output` de `FPDF` qui permet de rendre la génération visible.

Ci contre, la visualisation de ce code.



Création de QR Code

Si au début du projet, mon maître de stage avait la volonté de préserver les Qr Code actuels des véhicules, nous avons compris (lors de l'élaboration du cahier des charges) que les recréer complètement était plus simple et efficace. L'idée étant que depuis chaque véhicule, la régulation et le gestionnaire puissent accéder ou générer des Qr Codes relatif au véhicule sélectionné.

Pour mettre en place cette fonctionnalité, j'ai ajouté au projet la librairie phpqrcode.

Cet accès simplifié est signifié par un bouton possédant l'icône intitulé Qr Code de font awesome :



Une fois sélectionné, ce lien renvoie sur la page QrCodeConstruct.php. Voici le comportement typique de cette page pour le profil du gestionnaire :

```
1.1 session_start();
1.2 $accueil='QRCODE';
1.3 if ($_SESSION['fk_emploi'] == 4){
1.4     session_abort();
1.5     require __DIR__."/../vue/gestionnaire.php";
1.6 };
1.7 include('../phpqrcode/qrlib.php');
```

```

1.8 $tempDir = '../vue/media/qrCode/';
1.9 if(isset($_POST['vehiculeId'])){
1.10     $immatriculation = $_POST['vehiculeId'];
1.11 }else{
1.12     $immatriculation = $_SESSION['newVehiculeImmat'];
1.13 }
1.14
1.15 /***** GENERATION DU QR CODE ROULANT *****/
1.16
1.17 $codeContentsRoulant =
1.18 'http://ambul7.com/nouveauSanios/index.php?alias='.$immatriculation;
1.19 $fileNameRoulant = 'QrcodeIdRoulant'.$immatriculation.'.png';
1.20 $pngAbsolutePathRoulant = $tempDir.$fileNameRoulant;
1.21
1.22 // generating
1.23 if (!file_exists($pngAbsolutePathRoulant)) {
1.24     QRcode::png($codeContentsRoulant, $pngAbsolutePathRoulant, 1.25
QR_ECLEVEL_L, 3);
1.26 }
1.27
1.28 /***** GENERATION DU QR CODE ARS *****/
1.29
1.30 $codeContentsArs =
1.31 '

Ici la 1.1 à la 1.6 servent à définir le profil et afficher le header. La session est ouverte pour récupérer les données de Session, notamment le profil de l'utilisateur. Puis une fois que l'utilisateur est identifié, la session est stoppée. Elle est de nouveau ouverte à l'appel de gestionnaire.php. L'idée est de pouvoir ouvrir temporairement une session pour vérifier ses informations puisque son ouverture est indispensable dans gestionnaire.php.


```

La I.7 permet l'inclusion dans le fichier de la librairie et de ses méthodes ou fonctions. En I.8 Nous plaçons dans une variable nommée \$tempDir une chaîne de caractère qui correspond au chemin absolu vers un fichier destiné à récupérer les png des qr codes générés.

De la I.9 à la I.13, il s'agit d'une définition de la variable \$immatriculation. Son contenu change en fonction de l'arrivée utilisateur sur cette page (s'il arrive depuis la création de véhicule ou depuis le profil régule ou gestionnaire)

I.17 à I.20 Nous définissons des variables qui servent à définir le contenu du Qr Code, son chemin ou le nom que son image va porter. A chaque fois, l'immatriculation du véhicule est concaténée à celle du chemin. Enfin en I.24 nous créons le png du qr code.

Mais l'affichage de nouveau composant n'est pas encore complet. C'est la raison pour laquelle nous ajoutons en I.41 la vue qrCodeGenerator.php.

Cette dernière permet l'affichage des qr codes grace à :

```
<?php echo ''; ?>
```

C'est également grâce aux variables déterminées dans qrCodeConstruct.php qu'est généré le pdf puisque c'est cette vue qui contient le lien vers le pdf :

```
<a      target="_blank"      class="btn      btn-primary      px-4      py-2"
href=" ../vue/pdfGenerator.php?roulantpath=<?=$pngAbsolutePathRoulant?>&arspath=<?=$pngAbsolutePathArs?>&id=<?=$immatriculation?>"  >  Télécharger
les fichiers pdf </a>
```

Enfin, afin de ne pas surcharger les dossier d'images png, la page gestionnaireVehicules.php possède le code suivant :

```
if (file_exists('media/qrCode/QrcodeIdRoulant'.$vehicule['plaque'].'.png')) {
    unlink('media/qrCode/QrcodeIdRoulant'.$vehicule['plaque'].'.png');
    unlink('media/qrCode/QrcodeIdARS'.$vehicule['plaque'].'.png');
}
```

De cette manière, à chaque fois que l'utilisateur retourne sur la vue de tous les véhicules, tous les png des Qr codes qui ont été générés sont détruits. Ils seront recréés lors d'un prochain appel à qrCodeConstruct.php.

CONCLUSION

Le développement de Sanios s'est avéré être un défi pour moi. Appréhender le fonctionnement attendu et évoluer dans un milieu qui ne m'était pas familier représentaient en début de projet mes appréhensions majeures. Si établir rapidement et clairement un cahier des charges complet m'a aidée à comprendre le projet dans son architecture, je me suis longtemps posée des questions sur l'angle le plus approprié à aborder en premier dans ma façon de le concevoir.

Cependant après les dix semaines allouées par le stage il était véritablement satisfaisant de voir que les estimations faites avec mon tuteur étaient respectées. C'était un projet en constante évolution et qui devait respecter à la fois une simplicité et une complexité. Il devait fonctionner correctement sous la pression d'un grand nombre de données mais devait également rester fluide au niveau du code pour permettre une maintenance et une adaptabilité efficaces.

Aujourd'hui, il est résolument tourné vers l'application web et permet de s'adapter aux besoins futurs de l'entreprise. Pour ma part, il était important d'arriver à ce résultat hybride entre le web et l'application puisque j'ai déjà fait les démarches pour me diriger vers la formation concepteurs, développeur d'applications de l'ENI. Sanios était pour moi une façon d'expérimenter, de découvrir et de perfectionner mes capacités.

A n n e x e s

CAHIER DES CHARGES

61

DÉMONSTRATION DES FORMULAIRES

63



CAHIER DES CHARGES

Interface web de Gestion et désinfection des véhicules sanitaires
Etoile Secours

CAHIER DES CHARGES	37
DÉFINITION DE SANIOS	38
ENJEUX ET OBJECTIFS	39
SPÉCIFICITÉS TECHNIQUES	39
CHARTRE GRAPHIQUE	40
APERÇUS DES CONTENUS	41
DIAGRAMMES D'ACTIVITÉ	48
VOCABULAIRE DES PROFILS :	51
CONCEPTION DE LA BASE DE DONNÉES	51
<i>Structure des tables</i>	51
<i>Vues Relationnelles.</i>	58
CONTRAINTES PAR PROFILS	59
VISUALISATION DU CRUD PAR PROFIL	62

DÉFINITION DE SANIOS

Utilisé depuis 2018, Sanios est une application web qui permet la gestion d'un parc automobile relatif aux véhicules de soin ou du milieu médical. C'est une application interne à l'entreprise Etoile Secours. Il permet aux personnels qualifiés de signifier des hygiène effectuées sur les véhicules, exporter des pdf destinés à l'ARS (Agence Régionale de la Santé) et créer des QR codes à afficher sur les véhicules.

ENJEUX ET OBJECTIFS

L'enjeu du commanditaire du projet est de mettre à jour cette application web en créant une nouvelle version de celle-ci. Les nouveaux besoins exprimés sont les suivants :

- Mettre à jour un visuel déprécié
- Faciliter la navigation par utilisateur
- Créer une navigation articulée autour de différents profils.
- Donner la possibilité de renseigner un entretien technique
- Donner la possibilité de voir l'ensembles des véhicules à un administrateur
- Donner la possibilité de voir les véhicules d'une société par société.
- Mettre à jour les QR Codes
- Simplifier l'arborescence du projet pour en faciliter la maintenance.
- Garder la capacité de renseigner une hygiène
- Garder la capacité d'exporter des pdf sur des profils dédiés uniquement

- Protéger des données sensibles et privées.
- Agréer et harmoniser des données éparses.
- Vérifier l'emplacement d'un véhicule

SPÉCIFICITÉS TECHNIQUES

Utilisation des librairies et framework ci après cités.

Visual Studio Code.

Github Desktop

PHP QR Code est une librairie LGPL pour générer du QR Code

export pdf : FPDF

Bootstrap framework

CHARTRE GRAPHIQUE

Le graphisme de cette nouvelle version de Sanios se doit de garder l'identité de la version existante tout en la modernisant. Il se doit également de rester cohérent avec les couleurs associées au milieu médical. Des tons bleus et pastels seront privilégiés. De plus, l'implémentation des couleurs de base de bootstrap sera prise en compte. Les classes :

- Primary : associée en majorité aux profils régulation, mécanicien, et roulant.
- Secondary : associée en majorité au profil gestionnaire.

Définition d'un logo pour Sanios rappelant l'ancien.

Version Sanios 2018 :



Nouvelle version 2022 :



Code couleurs.

Nom

code Hex

Démonstration

whiteBlue	#CBEAE7	
lightBlue	#a9d6f1	
blue	#566592	
darkBlue	#06123E	
lightGreen	#D6F48F	
lightYellow	#F4EC8F	
lightRed	#F49B8F	
lightOrange	#FFEEDD	
lightGrey	#dbdbdb	

Couleurs complémentaires : variables de couleurs Bootstrap.

Font Family : variable de font Bootstrap.

Une vidéo sera implémentée sur la page d'accueil de la Régulation et du Gestionnaire afin d'apporter du dynamisme et de la modernité. Elle sera fixe et adaptable.

Site de provenance : <https://www.pexels.com/>

Auteur de la vidéo : <https://www.pexels.com/fr-fr/@cottonbro/>

Autorisations : libre de droits, libre de modification.

APERÇUS DES CONTENUS

Après un rapide topo sur la direction visuelle que devait prendre Sanios, j'ai élaboré des maquettes. Certaines ont été approuvées, d'autres on demandé une révision. Voici l'ensemble de ce travail.

Régule :

Bonjour nomSociete

Button

Vehicules

Button

Button

search

Button

Alias	vehicule	Immatriculation	Désinfection approfondie	Désinfection Courante	Nettoyage Extérieur	Nettoyage Interieur
AA1	OPEL	FK352FR	✗	✗	✓	✗
Désinfection Approfondie Désinfection Courante Désinfection Approfondie Nettoyage Extérieur			N/A N/A 2022-05-04 N/A	N/A N/A BELLANGER N/A	Véhicule location : 	
AA2	OPEL	FK352FR	✓	✗	✓	✓

Ajouter Véhicule

Immatriculation

Immatriculation

marque

modele

type

Date de mise en circulation

date de mise en circulation

Date du controle technique

date du controle technique

Alias

ALIAS

Button

Retoqué.

- Un menu doit apparaître en haut
- Sur une seconde page, on ajoutera la partie ajouter un véhicule.
- Sur la page contenant les véhicules un code couleur sera nécessaire.
- Une visualisation des derniers entretiens serait souhaitable

Régule v.2:

Bonjour nomSociete

Button

Tous les véhicules modifier / créer un véhicule

une légende
une légende
une légende
une légende

une légende
une légende
une légende
une légende

Vehicules


tableau de bords marques chercher par alias Button

Alias	vehicule	Immatriculation	Désinfection approfondie	Désinfection Courante	Nettoyage Extérieur	Nettoyage Intérieur	Qr Code																											
AA1	OPEL	FK352FR	✗	✗	✓	✗	QR Code +																											
<p>HYGIENES</p> <table border="1"> <tr> <td>Désinfection Approfondie</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td>Désinfection Courante</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td>Désinfection Approfondie</td> <td>2022-05-04</td> <td>BELLANGER</td> </tr> <tr> <td>Nettoyage Extérieur</td> <td>N/A</td> <td>N/A</td> </tr> </table> <p>ENTRETIENS</p> <table border="1"> <tr> <td>vidange</td> <td>date</td> <td>personnel</td> </tr> <tr> <td>parebrise</td> <td>date</td> <td>personnel</td> </tr> <tr> <td>carrosserie</td> <td>date</td> <td>personnel</td> </tr> <tr> <td>type entretien</td> <td>date</td> <td>personnel</td> </tr> <tr> <td>type entretien</td> <td>date</td> <td>personnel</td> </tr> </table>								Désinfection Approfondie	N/A	N/A	Désinfection Courante	N/A	N/A	Désinfection Approfondie	2022-05-04	BELLANGER	Nettoyage Extérieur	N/A	N/A	vidange	date	personnel	parebrise	date	personnel	carrosserie	date	personnel	type entretien	date	personnel	type entretien	date	personnel
Désinfection Approfondie	N/A	N/A																																
Désinfection Courante	N/A	N/A																																
Désinfection Approfondie	2022-05-04	BELLANGER																																
Nettoyage Extérieur	N/A	N/A																																
vidange	date	personnel																																
parebrise	date	personnel																																
carrosserie	date	personnel																																
type entretien	date	personnel																																
type entretien	date	personnel																																
<p>Véhicule location Date de controle technique</p>																																		
AA2	OPEL	FK352FR	✓	✗	✓	✓	QR Code +																											
AA2	OPEL	FK352FR	✓	✗	✓	✓	QR Code +																											


Bonjour nomSociete

Button

Tous les véhicules modifier / créer un véhicule qr code



IMMATRICULATION
AMBULANCIERS



IMMATRICULATION
ARS

Télécharger le pdf

Bonjour nomSociete

Button

Tous les véhicules modifier / créer un véhicule

Modifier un véhicule

nom - immatriculation - marque - modele

modifier

Créer un véhicule

marque

créer

Formulaire création (version complète) :

Créer un véhicule

Ambulances SOCIETE

Marque

Modèle du véhicule

Type du véhicule

Alias du véhicule

Immatriculation du véhicule

Kilométrage

Inventaire position

Date de mise en circulation

Date du controle Technique

Button

Formulaire Modification (version complète) :

MODIFIER LE VEHICULE - AT38

Société du véhicule

Marque du véhicule

Modele du véhicule

Type du véhicule

Plaque d'immatriculation

Inventaire Position

Date de mise en circulation

Cession du véhicule

Je valide mes modifications

Gestionnaire :

* Bonjour PAC

Me deconnecter

Voir tous les véhicules

Modifier / Créer un véhicule

Voir les véhicules cédés

Voir le personnel

☐ Véhicule conforme
☐ Véhicule localisé hors de votre société
☐ Date du controle technique à venir (sous 15 Jours)
☐ Date du controle technique dépassé

☐ Entretien effectué
☐ Aucune donnée valide
☐ Entretien à effectuer en urgence

Véhicules - Ambulances ASUR

Tableau de Bord

sociétés

Chercher par Alias

Chercher

Alias	Véhicule	Immatriculation	société	QRcode
AA11	OPEL	FK113TJ	Ambulances ASUR	+
AA12	FIAT	FV207WM	Ambulances ASUR	+
AA13	FIAT	FX838PT	Ambulances ASUR	+
AT33	SKODA	DF074XF	Ambulances ASUR	+
AT35	PEUGEOT	EC549CL	Ambulances ASUR	+
AT37	RENAULT	FD066RM	Ambulances ASUR	+

Bonjour PAC

Me déconnecter

Voir tous les véhicules

Modifier / Créer un véhicule

Voir les véhicules cédés

Voir le personnel

Cessions de véhicules

id	Véhicule	Immatriculation	société
TEST	DACIA	TE222SF	Ambulances ASUR

Hygienes

Désinfection approfondie

N/A

N/A

Désinfection courante

N/A

N/A

Nettoyage Extérieur

N/A

N/A

Nettoyage Intérieur

N/A

N/A

Entretiens

Vidange

N/A

N/A

N/A

Distribution

N/A

N/A

N/A

Carrosserie

N/A

N/A

N/A

Pare Brise

N/A

N/A

N/A

Plaquette de Freins

N/A

N/A

N/A

Disque de Freins

N/A

N/A

N/A

Le véhicule à été cédé

!

Date du prochain controle Technique 2026-06-16

TEST	DACIA	TE222SF	Ambulances ASUR
------	-------	---------	-----------------

Bonjour PAC

Me déconnecter

Voir tous les véhicules

Modifier / Créer un véhicule

Voir les véhicules cédés

Voir le personnel

Personnel

Tableau de Bord

Personnel

Sociétés

Rechercher par Nom

Chercher

Profil	Nom	Prenom	Societe
AMBULANCIER	BELANGER	mon-prenom	Ambulances ASUR
MECANICIEN	RICHARD	maxime	Ambulances ASUR
REGULATION	PETROW	patrice	Ambulances ASUR
REGULATION	BASTIAN	marie-christine	Ambulances ASUR

Ambulanciers :

The first screen shows a login page with the text 'Bonjour nom' and a 'Me deconnecter' button. The second screen shows a form for entering vehicle information: 'Véhicule : ALIAS', 'Modèle : Nom', 'Immatriculation : PLAQUE', and 'Date du jour : date'. Below this is a section titled 'Type d'Hygiène' with four radio button options: 'Type 1', 'Type 2', 'Type 3', and 'Type 4', and a 'Je valide' button. The third screen shows a confirmation message: 'Merci d'avoir validé votre Début de conduite' with a large green checkmark. The fourth screen shows the same vehicle information form as the second screen, but with 'Type 3' selected.

Mécaniciens :

The first screen shows a login page for a mechanic with the text 'Bonjour RICHARD' and 'véhicule FK113TJ'. It has a 'Me deconnecter' button. The second screen shows a form for entering vehicle information: 'Véhicule : FK113TJ', 'Immatriculation : FK113TJ', 'Marque : OPEL', and 'kilometrage : 221122'. Below this is a section titled 'Mettre à jour le kilometrage du véhicule' with a label 'Relevé du kilometrage' and a text input field. There is a 'Je valide' button at the bottom.

Véhicule : FK113TJ

OPEL - Vivaro

Renseigner un entretien :

Vidange

Distribution

Carrosserie

Pare brise

Plaquettes de freins


Disque de freins

Relevé du kilometrage

Ajouter une remarque

Je valide

ARS:



Identité du véhicule

Ambulances ASUR

numero véhicule : AA11

marque véhicule : OPEL

immatriculation : FK113TJ

date du jour : 20-07-2022

[voir le pdf](#)

Hygiene du véhicule


Date de l'hygiène	Type d'hygiène	Personnel
2022-06-23	Nettoyage Extérieur	BELANGER
2022-06-23	Désinfection courante	BELANGER
2022-06-23	Désinfection approfondie	BELANGER
2022-06-27	Désinfection courante	BELANGER

1 / 1

100%

+

+



Desinfection des vehicules

Semaine : 29 - Année : 2022

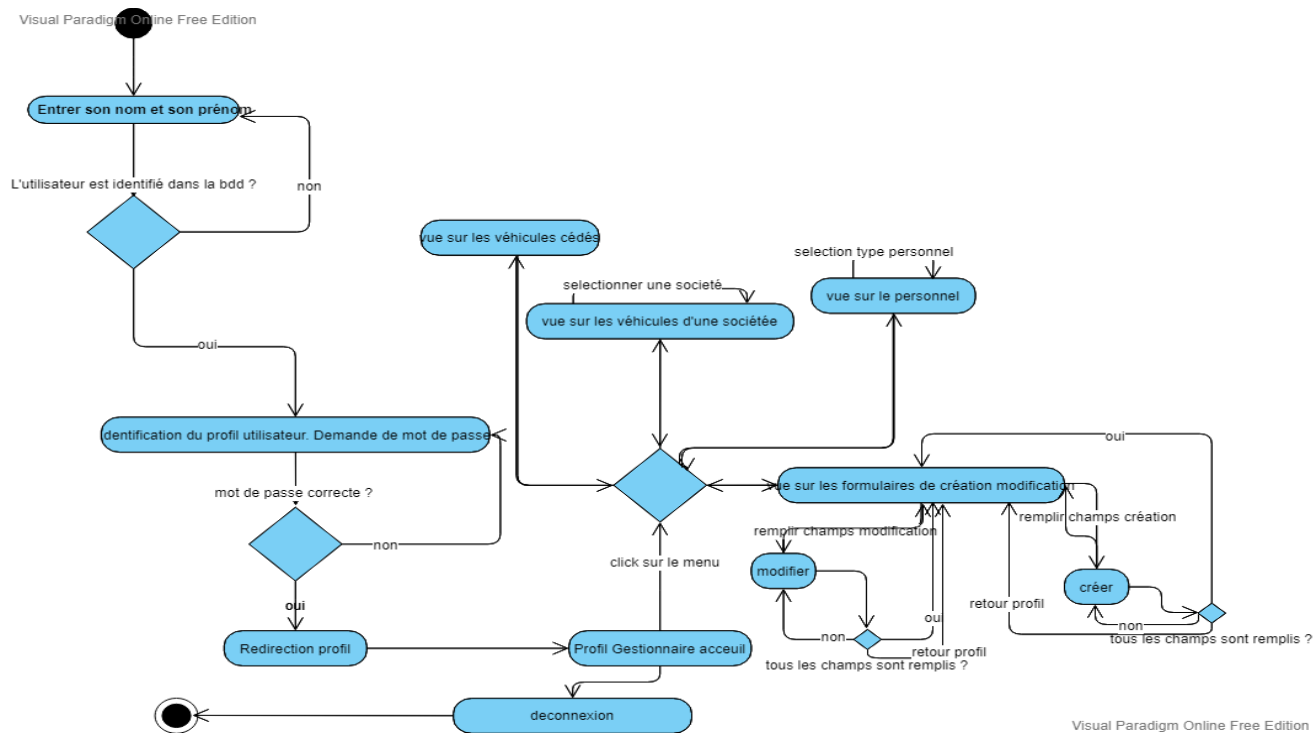
Ambulances ASUR - vehicule : FK113TJ

DATE	TYPE DE DESINFECTION	NOM
2022-06-23	Nettoyage Extérieur	BELANGER
2022-06-23	Désinfection courante	BELANGER
2022-06-23	Désinfection approfondie	BELANGER
2022-06-27	Désinfection courante	BELANGER

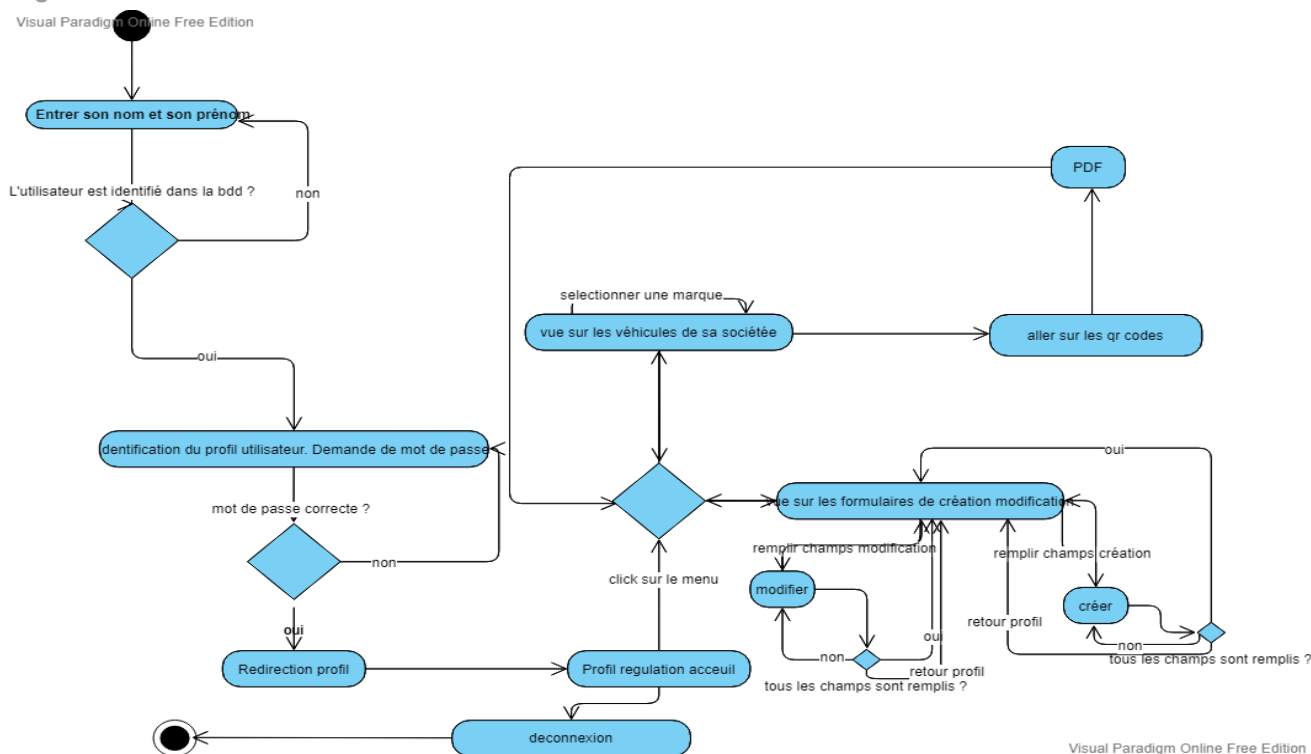
DIAGRAMMES D'ACTIVITÉ

Diagramme d'activités par profil :

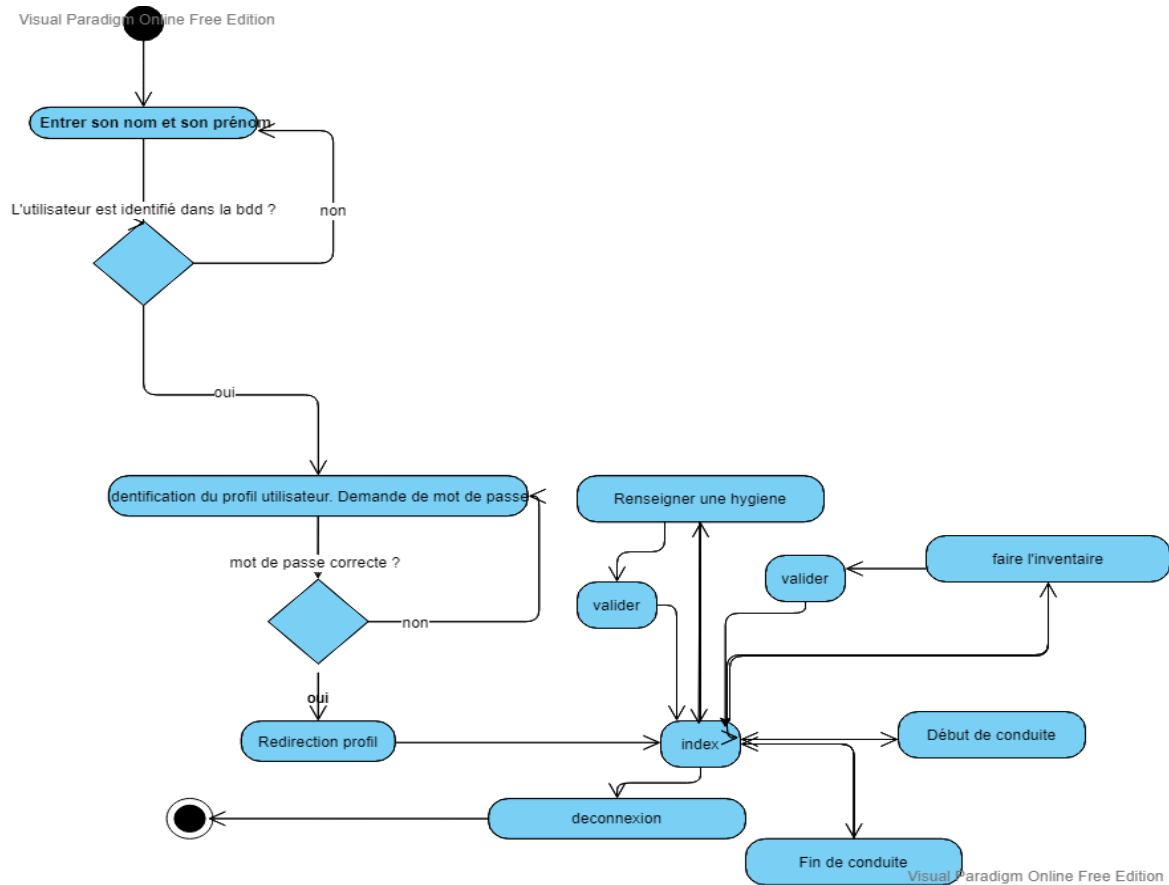
Gestionnaire :



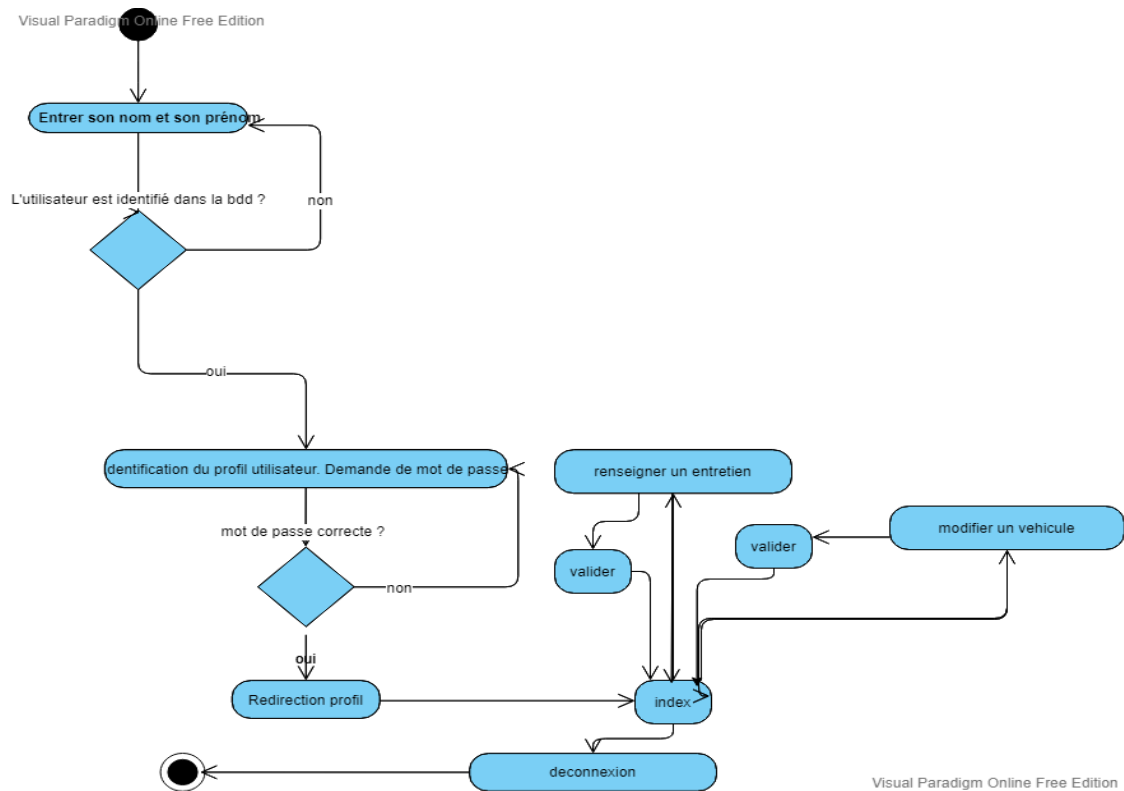
Regulation:



Ambulancier:



Mecanicien:



VOCABULAIRE DES PROFILS :

L e x i q u e	
V.S.L :	Véhicule Sanitaire Léger.
Ambulance :	Ambulances de classe C
ASSU :	Ambulances contenant le matériel nécessaire aux soins d'urgences.
Roulants / Ambulanciers :	Conducteurs des véhicules incluant les V.S.L, les Ambulances et les ambulances de type ASSU.
ARS :	Autorité Régionale de la Santé.
Régulation :	Ambulancier dont le travail principal est de répondre au téléphone et d'organiser les déplacements de ses collègues Roulants.
Mécanicien :	Effectue des entretiens sur les véhicules de tous types.
Gestionnaire :	Administrateur du parc automobile.

CONCEPTION DE LA BASE DE DONNÉES

Structure des tables

sanios_marque

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	Commentaires	MIME
id_marque	int(9)		Non		auto_increment			
marque_nom	varchar(20)		Non					

sanios_modele

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	Commentaires	MIME
id_modele	int(9)		Non		auto_increment			
fk_marque	int(11)		Non			-> sanios_marque.id_marque ON UPDATE CASCADE ON DELETE CASCADE		
modele_nom	varchar(20)		Non					

sanios_vehicule

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_vehicule	int(9)		Non		auto_increment			
plaque	varchar(8)		Non					
fk_marque	int(2)		Non			-> sanios_marque.id_marque ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_modele	int(2)		Non			-> sanios_modele.id_modele ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_type	int(4)		Non	22		-> sanios_type.id_type ON UPDATE CASCADE ON DELETE CASCADE		
fk_societe	int(99)		Non			-> sanios_societe.id_societe ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_inventaire_position	int(99)		Non			-> sanios_societe.id_societe ON UPDATE CASCADE ON DELETE CASCADE		
date_mise_circulation	date		Non					
date_controle_technique	date		Non					
kilometrage	int(6)		Non					
cession	tinyint(1)		Non					
ancien_id	varchar(6)		Non					

sanios_type

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_type	int(11)		Non		auto_increment			
type_nom	varchar(30)		Non					

sanios_societe

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_societe	int(9)		Non		auto_increment			
societe_nom	varchar(40)		Non					

sanios_utilisateur

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_user	int(11)		Non		auto_increment			
nom	varchar(30)		Non					
prenom	varchar(30)		Non					
mdp	varchar(255)		Non					
fk_societe	int(99)		Non			-> sanios_societe.id_societe ON UPDATE CASCADE ON DELETE CASCADE		
fk_emploi	int(99)		Non			-> sanios_emploi.id_emploi ON UPDATE RESTRICT ON DELETE RESTRICT		
statut	tinyint(1)		Non					
date_naissance	date		Non					

sanios_emploi

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_emploi	int(9)		Non		auto_increment			
emploi_nom	varchar(35)		Non					

sanios_hygiene

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_entretien	int(11)		Non		auto_increment			
fk_vehicule	int(11)		Non			-> sanios_vehicule.id_vehicule ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_user	int(2)		Non			-> sanios_utilisateur.id_user ON UPDATE RESTRICT ON DELETE RESTRICT		
date	date		Non					
heure	datetime		Non	CURRENT_TIMESTAMP				
fk_user_societe	int(11)		Non			-> sanios_societe.id_societe ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_vehicule_societe	int(11)		Non			-> sanios_societe.id_societe ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_type	int(11)		Non			-> sanios_hygiene_type.id_hygiene_type ON UPDATE CASCADE ON DELETE CASCADE		

sanios_hygiene_type

Colonne	Type	Attributs	Nu	Meur par défaut	Extra	Est relié à	Commentaires	MIME
id_hygiene_type	int(11)		Non		auto_increment			
nom	varchar(30)		Non					

sanios_entretien

Colonne	Type	Attributs	Nu	Meur par défaut	Extra	Est relié à	Commentaires	MIME
id_mecanique	int(100)		Non		auto_increment			
fk_vehicule	int(11)		Non			-> sanios_vehicule.id_vehicule ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_user	int(2)		Non			-> sanios_utilisateur.id_user ON UPDATE RESTRICT ON DELETE RESTRICT		
date	date		Non					
heure	datetime		Non					
fk_user_societe	int(11)		Non			-> sanios_utilisateur.fk_societe ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_vehicule_societe	int(11)		Non			-> sanios_societe.id_societe ON UPDATE RESTRICT ON DELETE RESTRICT		
fk_kms	int(6)		Non					
fk_type	int(3)		Non			-> sanios_entretien_type.id_entretien_type ON UPDATE CASCADE ON DELETE CASCADE		
autre	varchar(255)		Non					
externe	varchar(255)		Non					

sanios_entretien_type

Colonne	Type	Attributs	Nu	Meur par défaut	Extra	Est relié à	Commentaires	MIME
id_entretien_type	int(11)		Non		auto_increment			
nom	varchar(40)		Non					
alerte	int(5)		Non					

sanios_liste_inventaire_ambu

Colonne	Type	Attributs	Nu	Meur par défaut	Extra	Est relié à	Commentaires	MIME
id_produit	int(3)		Non					
nom	varchar(30)		Non					
quantite	int(2)		Non					

sanios_inventaire_ambu

Colonne	Type	Attributs	Nu	Meur par défaut	Extra	Est relié à	Commentaires	MIME
id_inventaire	int(4)		Non		auto_increment			
date_inventaire	date		Non					
fk_id_vehicule	int(4)		Non			-> sanios_vehicule.id_vehicule ON UPDATE CASCADE ON DELETE CASCADE		
fk_id_user	int(5)		Non			-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
fk_id_user2	int(5)		Oui	NULL		-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
10P	int(11)		Non					
10D	date		Oui	NULL				
20P	int(2)		Non					
20D	date		Oui	NULL				
30P	int(2)		Non					
30D	date		Oui	NULL				
40P	int(2)		Non					
40D	date		Oui	NULL				
50P	int(2)		Non					
50D	date		Oui	NULL				
60P	int(2)		Non					
60D	date		Oui	NULL				
70P	int(2)		Non					
70D	date		Oui	NULL				
80P	int(2)		Non					
80D	date		Oui	NULL				
90P	int(2)		Non					
90D	date		Oui	NULL				
100P	int(2)		Non					
100D	date		Oui	NULL				
110P	int(2)		Non					
110D	date		Oui	NULL				
120P	int(2)		Non					
120D	date		Oui	NULL				
130P	int(2)		Non					
130D	date		Oui	NULL				
140P	int(2)		Non					

140D	date		Oui	NULL				
150P	int(2)		Non					
150D	date		Oui	NULL				
160P	int(2)		Non					
160D	date		Oui	NULL				
170P	int(2)		Non					
170D	date		Oui	NULL				
180P	int(2)		Non					
180D	date		Oui	NULL				
190P	int(2)		Non					
190D	date		Oui	NULL				
200P	int(2)		Non					
200D	date		Oui	NULL				
210P	int(2)		Non					
210D	date		Oui	NULL				
220P	int(2)		Non					
220D	date		Oui	NULL				
230P	int(2)		Non					
230D	date		Oui	NULL				
240P	int(2)		Non					
240D	date		Oui	NULL				
250P	int(2)		Non					
250D	date		Oui	NULL				
260P	int(2)		Non					
260D	date		Oui	NULL				
270P	int(2)		Non					
270D	date		Oui	NULL				
280P	int(2)		Non					
280D	date		Oui	NULL				
290P	int(2)		Non					
290D	date		Oui	NULL				
300P	int(2)		Non					
300D	date		Oui	NULL				
310P	int(2)		Non					
310D	date		Oui	NULL				
320P	int(2)		Non					
320D	date		Oui	NULL				
330P	int(2)		Non					
330D	date		Oui	NULL				
340P	int(2)		Non					
340D	date		Oui	NULL				
350P	int(2)		Non					
350D	date		Oui	NULL				
360P	int(2)		Non					
360D	date		Oui	NULL				
370P	int(2)		Non					
370D	date		Oui	NULL				
380P	int(2)		Non					

sanios_liste_inventaire_assu

Colonne	Type	Attributs	Nu	Meur par défaut	Extra	Est relié à	Commentaires	MIME
id_produit	int(3)		Non					
nom	varchar(30)		Non					
quantite	int(2)		Non					

sanios_inventaire_assu

Colonne	Type	Attributs	Num	Valeur par défaut	Extra	Est relié à	Commentaires	MIME
id_inventaire	int(4)		Non		auto_increment			
date_inventaire	date		Non					
fk_id_vehicule	int(4)		Non			-> sanios_vehicule.id_vehicule ON UPDATE CASCADE ON DELETE CASCADE		
fk_id_user	int(5)		Non			-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
fk_id_user2	int(5)		Oui	NULL		-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
10P	int(11)		Non					
10D	date		Oui	NULL				
20P	int(2)		Non					
20D	date		Oui	NULL				
30P	int(2)		Non					
30D	date		Oui	NULL				
40P	int(2)		Non					
40D	date		Oui	NULL				
50P	int(2)		Non					
50D	date		Oui	NULL				
60P	int(2)		Non					
60D	date		Oui	NULL				
70P	int(2)		Non					
70D	date		Oui	NULL				
80P	int(2)		Non					
80D	date		Oui	NULL				
90P	int(2)		Non					
90D	date		Oui	NULL				
100P	int(2)		Non					
100D	date		Oui	NULL				
110P	int(2)		Non					
110D	date		Oui	NULL				
120P	int(2)		Non					
120D	date		Oui	NULL				
130P	int(2)		Non					
130D	date		Oui	NULL				
140P	int(2)		Non					

140D	date		Oui	NULL				
150P	int(2)		Non					
150D	date		Oui	NULL				
160P	int(2)		Non					
160D	date		Oui	NULL				
170P	int(2)		Non					
170D	date		Oui	NULL				
180P	int(2)		Non					
180D	date		Oui	NULL				
190P	int(2)		Non					
190D	date		Oui	NULL				
200P	int(2)		Non					
200D	date		Oui	NULL				
210P	int(2)		Non					
210D	date		Oui	NULL				
220P	int(2)		Non					
220D	date		Oui	NULL				
230P	int(2)		Non					
230D	date		Oui	NULL				
240P	int(2)		Non					
240D	date		Oui	NULL				
250P	int(2)		Non					
250D	date		Oui	NULL				
260P	int(2)		Non					
260D	date		Oui	NULL				
270P	int(2)		Non					
270D	date		Oui	NULL				
280P	int(2)		Non					
280D	date		Oui	NULL				
290P	int(2)		Non					
290D	date		Oui	NULL				
300P	int(2)		Non					
300D	date		Oui	NULL				
310P	int(2)		Non					
310D	date		Oui	NULL				
320P	int(2)		Non					
320D	date		Oui	NULL				
330P	int(2)		Non					
330D	date		Oui	NULL				
340P	int(2)		Non					
340D	date		Oui	NULL				
350P	int(2)		Non					
350D	date		Oui	NULL				
360P	int(2)		Non					
360D	date		Oui	NULL				
370P	int(2)		Non					
370D	date		Oui	NULL				
380P	int(2)		Non					

380D	date		Oui	NULL				
390P	int(2)		Non					
390D	date		Oui	NULL				
400P	int(2)		Non					
400D	date		Oui	NULL				
410P	int(2)		Non					
410D	date		Oui	NULL				
420P	int(2)		Non					
420D	date		Oui	NULL				
430P	int(2)		Non					
430D	date		Oui	NULL				
440P	int(2)		Non					
440D	date		Oui	NULL				
450P	int(2)		Non					
450D	date		Oui	NULL				
460P	int(2)		Non					
460D	date		Oui	NULL				
470P	int(2)		Non					
470D	date		Oui	NULL				
480P	int(2)		Non					
480D	date		Oui	NULL				
490P	int(2)		Non					
490D	date		Oui	NULL				
500P	int(2)		Non					
500D	date		Oui	NULL				
510P	int(2)		Non					
510D	date		Oui	NULL				
520P	int(2)		Non					
520D	date		Oui	NULL				
530P	int(2)		Non					
530D	date		Oui	NULL				
540P	int(2)		Non					
540D	varchar(4)		Oui	NULL				
550P	int(2)		Non					
550D	varchar(4)		Oui	NULL				

sanios_liste_inventaire_vsl

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	Commentaires	MIME
id produit	int(3)		Non					
nom	varchar(130)		Non					
quantite	int(2)		Non					

sanios_inventaire_vsl

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id_inventaire	int(4)		Non		auto_increment			
date_inventaire	date		Non					
fk_id_vehicule	int(4)		Non			-> sanios_vehicule.id_vehicule ON UPDATE CASCADE ON DELETE CASCADE		
fk_id_user	int(5)		Non			-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
fk_id_user2	int(5)		Oui	NULL		-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
10P	int(11)		Non					
10D	date		Non					
20P	int(2)		Non					
20D	date		Non					
30P	int(2)		Non					
30D	date		Non					
40P	int(2)		Non					
40D	date		Non					
50P	int(2)		Non					
50D	date		Non					
60P	int(2)		Non					
62D	date		Non					
70P	int(2)		Non					
70D	date		Non					
80P	int(2)		Non					
80D	date		Non					
90P	int(2)		Non					
90D	date		Non					
100P	int(2)		Non					
100D	date		Non					
110P	int(2)		Non					
110D	date		Non					
120P	int(2)		Non					
120D	date		Non					
130P	int(2)		Non					
130D	date		Non					
140P	int(2)		Non					

140D	date		Non					
150P	int(2)		Non					
150D	date		Non					
160P	int(2)		Non					
160D	date		Non					
170P	int(2)		Non					
170D	date		Non					
180P	int(2)		Non					
180D	date		Non					
190P	int(2)		Non					
190D	date		Non					
200P	int(2)		Non					
200D	date		Non					
210P	int(2)		Non					
210D	date		Non					
220P	int(2)		Non					
220D	date		Non					
230P	int(2)		Non					
230D	date		Non					
240P	int(2)		Non					
240D	date		Non					
250P	int(2)		Non					
250D	date		Non					
260P	int(2)		Non					
260D	date		Non					
270P	int(2)		Non					
270D	date		Non					
280P	int(2)		Non					
280D	date		Non					
290P	int(2)		Non					
290D	date		Non					
300P	int(2)		Non					
300D	date		Non					
310P	int(2)		Non					
310D	date		Non					
320P	int(2)		Non					
330P	int(2)		Non					

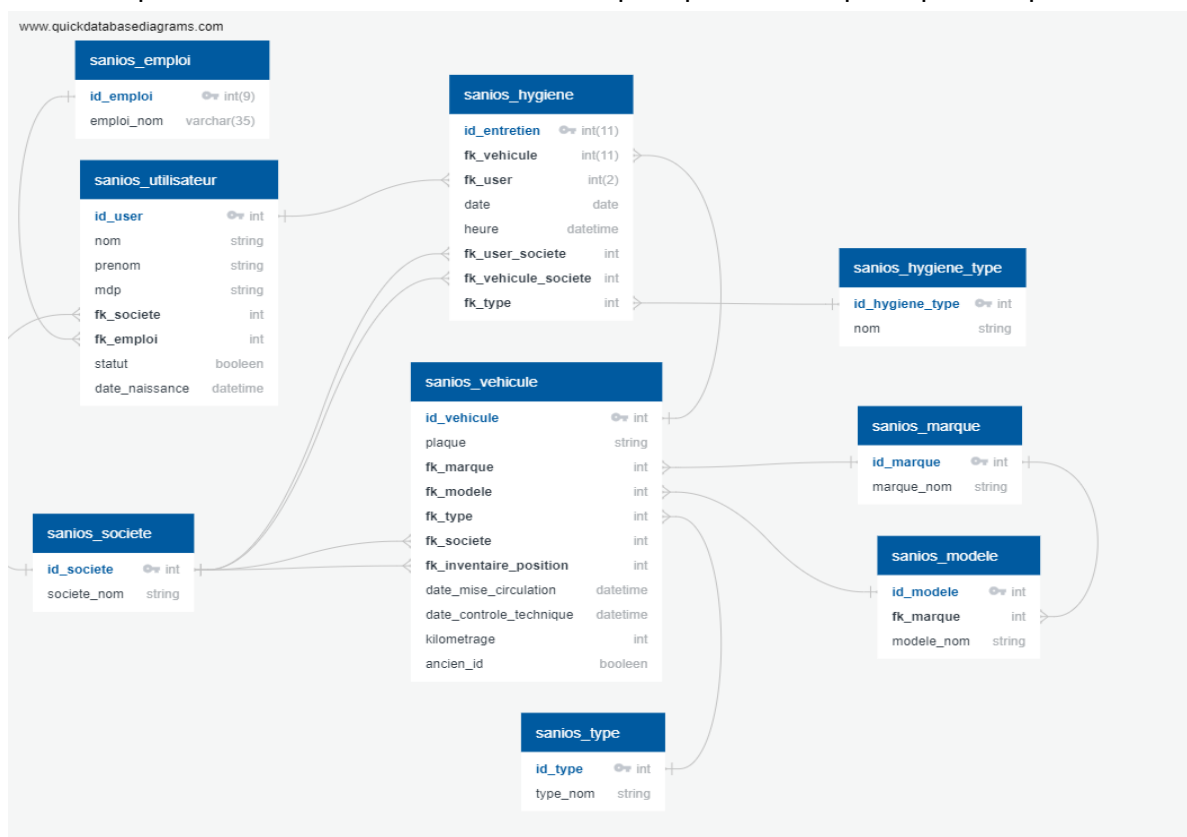
sanios_conduite

Colonne	Type	Attributs	Nu	leur par défaut	Extra	Est relié à	Commentaires	MIME
id	int(11)		Non		auto_increment			
fk_user	int(4)		Non			-> sanios_utilisateur.id_user ON UPDATE CASCADE ON DELETE CASCADE		
fk_vehicule	int(4)		Non			-> sanios_vehicule.id_vehicule ON UPDATE CASCADE ON DELETE CASCADE		
date	datetime		Non	CURRENT_TIMESTAMP				
conduite	tinyint(1)		Non					

Vues Relationnelles.

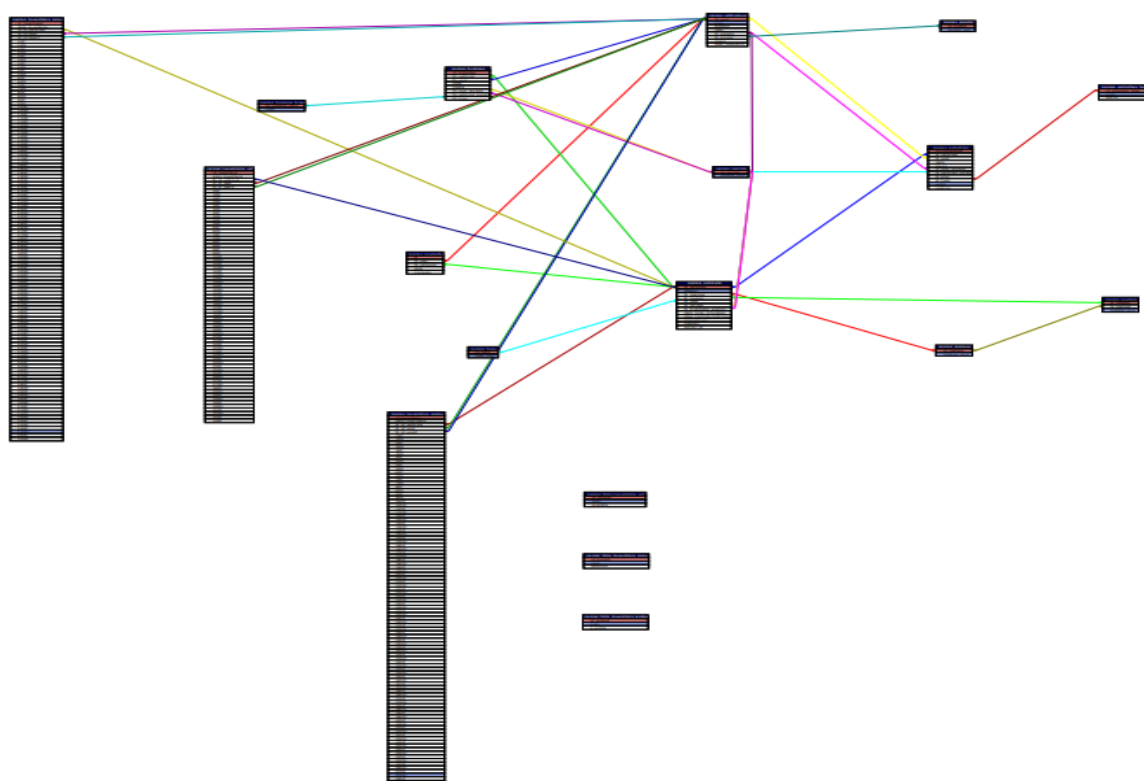
- Réorganiser les 26 bases de données existantes en une unique :

Répond au besoin d'unifier les sociétés pour pouvoir comparer plus simplement les données.



Ajout de nouvelles fonctionnalités dans cette nouvelle base de donnée :

Aperçu du nouveau schéma relationnel :



CONTRAINTES PAR PROFILS

En tant que (profil)

ROULANT / AMBULANCIER

Le Roulant scanne le QR CODE qui lui est relatif. Il arrive sur une interface :

- Il doit renseigner son nom, son prénom puis sa date de naissance
 - contraintes : le nom sera consulté dans une table nommée sanios_utilisateur de la base de données sanios. Les espaces contenus dans les noms seront transformés en '-'. Les accents contenus seront transformés en leur équivalence sans accents (ainsi "é" devient "e"). Les nom seront transformés en UPPERCASE
 - contraintes : le prénom sera consulté dans une table nommée sanios_utilisateur de la bdd sanios. Les espaces contenus dans les noms seront transformés en '-'. Les accents contenus seront transformés en leur équivalent sans accents (ainsi "é" devient "e"). Les prénoms seront transformés en LOWERCASE.

- Si et seulement si le nom est reconnu dans la base de donnée, il est redirigé vers un nouveau formulaire. Il doit alors rentrer sa date de naissance. Si elle est erronée un message apparaît.
 - contrainte : si un l'utilisateur 'ambulancier/roulant' tente une connexion sans scan de qr code, il est bloqué au niveau de la date de naissance.
- Il doit renseigner le type d'hygiène qu'il a effectué. (Nettoyage ou Désinfection)
- Il ne peut rentrer qu'une seule hygiène à la fois.
 - contrainte : dans la table nommée sanios_hygiene de la database sanios, doit figurer l'heure et la date , l'id du véhicule, ainsi que le nom de l'utilisateur qui viens d'indiquer une hygiène sur un véhicule.
- Il doit pouvoir faire une liste des éléments présents dans le véhicule (masque etc...)
- L'heure, la date , le nom de l'utilisateur et l'id du véhicules ainsi que son type doivent être ajoutés dans sanios_inventaire_(typevehicule)

En tant que (profil)

ARS

L'ARS scanne le QR CODE qui lui est relatif. Il arrive sur une interface :

- Peut consulter toutes les opérations d'hygiène sur 5 semaines. (du véhicule)
- Peut consulter ces données en PDF. Il peut aussi télécharger ce PDF.

En tant que (profil)

RÉGULE

- Accès à tous les véhicules de sa société.
- Accès pour un véhicule à ses données d'hygiène et d'entretien.
- Accès pour un véhicule à la date de son prochain controle technique
- Peut voir si un véhicule n'est pas localisé dans sa société. Si c'est le cas, il peut voir ou est le véhicule.
- Peut créer un véhicule (create sur sanios_vehicule)
 - A la création, il doit remplir les champs : plaque, ancien_id (qui lui est présenté sous le nom "alias"). Il doit choisir la marque et le modèle du véhicule.
 - contrainte : les modèles proposés doivent appartenir à une marque.
 - Il doit pouvoir ajouter une date de mise en circulation et une date de contrôle technique. Il doit également

► Va recevoir des alerte :

- Alerte limite kilométrage avant cession du véhicule.
 - 300 000 Km pour véhicules BM, Renault, Skoda, etc ...
 - 500 000 Km pour véhicules Toyota
- Alerte limite kilométrage avant vidange.
 - 10 000 Km
- Alerte limite temps et/ou kilométrage avant distribution.
 - 100 000 Km
 - 5 ans
 -

En tant que (profil)

MÉCANICIEN

Il scanne le QR CODE relatif aux ambulanciers et arrive sur l'interface. Un bouton consacré lui propose de s'identifier en tant que mécanicien. Cette action l'amène à une interface réservée pour son profil.

- Peut renseigner le type d'entretien effectué.
- Récupération de la date et de l'heure lors de la validation de saisie d'un entretien.
- Il peut mettre à jour des véhicules (nombre de kilomètres uniquement)

En tant que (profil)

GESTIONNAIRE DE PARC AUTOMOBILE

- Peut consulter la la position d'un véhicule.
- Peut consulter la liste de tous les véhicules par société.
- Peut consulter la liste des véhicules présents dans une société.

La différence entre la liste des véhicules par société et celle des véhicules présents dans la société réside dans le fait qu'un véhicule peut être dans une société à laquelle il n'est pas affilié. Permet de vérifier la position des véhicules.

- Peut consulter combien d'hygiène ont été faites par une personne.
- Accès pour un véhicule à ses données d'hygiène et d'entretien.
- Peut consulter la position de ses véhicules. (consulte les mêmes choses que le profil régule)
- Peut créer un véhicule (créer les mêmes choses que le profil régule)
- Il peut consulter l'historique de la mécanique véhicule. (accès au même tri que le profil mécanicien)
- Peut consulter toutes les interventions pour un véhicule (entretien et/ou hygiène)
- Va recevoir des alerte :

Alerte limite kilomètre avant cession du véhicule.

- 300 000 Km pour véhicules BM, Renault, Skoda, etc ...
- 500 000 Km pour véhicules Toyota

Alerte limite kilomètre avant vidange.

- 10 000 Km

Alerte limite temps et/ou kilomètre avant distribution.

- 100 000 Km
- 5 ans

....

► Peut modifier la cession d'un véhicule.

► Peut changer l'affectation d'une société.

► Alerte controle technique : si date controle technique existe alors date+2Y-15jours

si date controle technique existe pas alors date+4Y-15J

► Peut sortir des statistiques sur le travail des mécaniciens (accès à sanios_entretien)

VISUALISATION DU CRUD PAR PROFIL

ROULANT	
ACTION CRUD	TABLE AFFECTÉE
READ	sanios_vehicule (id_vehicule)
CREATE	sanios_hygiene

ARS	
ACTION CRUD	TABLE AFFECTÉE
READ	sanios_vehicule (id_vehicule)
READ	sanios_hygiene (nom) constraint : date du jour uniquement

RÉGULE	
ACTION CRUD	TABLE AFFECTÉE
READ	sanios_vehicule (par fk_societe)
READ	sanios_hygiene (par fk_societe)
READ	sanios_entretien (par fk_societe)

CREATE	sanios_vehicule
--------	-----------------

MÉCANICIEN	
ACTION CRUD	TABLE AFFECTÉE
READ	sanios_utilisateur (nom, mdp)
READ	sanios_vehicule (id_vehicule)
CREATE	sanios_entretien
UPDATE	sanios_vehicule (kilometrage, fk_societe)
READ	sanios_entretien

GESTIONNAIRE PARC AUTO	
ACTION CRUD	TABLE AFFECTÉE
READ	sanios_vehicule
READ	sanios_utilisateur
READ	sanios_hygiene
READ	sanios_entretien
UPDATE	sanios_vehicule (cession , date_controle_technique, fk_societe)
CREATE	sanios_vehicule

DÉMONSTRATION DES FORMULAIRES

Contenu de formulaire.php (Modifie ou créer un véhicule)

```
<?php

(!isset($_SESSION)) ? session_start() : session_abort();

if($_SESSION['fk_emploi'] != 4 && $_SESSION['fk_emploi'] != 3 ) {
    header('location: ../index.php');
}

if($_SESSION['fk_emploi'] == 4){
    (!isset($_SESSION)) ? session_start() : session_abort();
    $accueil = 1;
    $colorBtn = "secondary";
    $returnProfil = "gestionnaireForm.php";
    require __DIR__.'./gestionnaire.php';
}

if ($_SESSION['fk_emploi'] == 3) {
    (!isset($_SESSION)) ? session_start() : session_abort();
    $accueil = 1;
    $colorBtn = "primary";
    $returnProfil = "reguleForm.php";
    require __DIR__.'./regule.php';
}

require ('../vuemodel/formulaireVueModel.php');
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css
">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.c
ss">
<link rel="stylesheet" href="media/css/asset.css">
<title>Véhicules</title>
</head>

<body>

<div class="card m-4">

    <?php if (isset($_POST['modify'])) :

        $currentAlias = $_POST['alias']; ?>

    <div class="card-header pt-4 text-<?=$colorBtn?>">
        <h1> MODIFIER LE VEHICULE - <?=$currentAlias ?> </h1>
    </div>

    <div class="card-body pt-4">

        <div class="container">

            <form method="POST" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
                <div class="row pt-4 pb-3 ">
                    <?php if($_SESSION['fk_emploi'] == 4): ?>
                        <div class="col-lg-3 col-md-6">
                            <label for="societe">Société du véhicule</label>
                            <select class="form-select" name="societe" id="">
                                <option selected>
<?=$currentVehicule['societe_nom'] ?></option>
                                <?php foreach ($societes as $societe) : ?>
                                    <option value="<?=$societe['societe_nom'];
?>">

                                    <?=$societe['societe_nom']; ?>
                                </option>
                                <?php endforeach; ?>
                            </select>
                        </div>

```

```

        <?php else : ?>
            <h2><?=$currentVehicule['societe_nom'] ?></h2>
        <?php endif; ?>
    <div class="col-lg-3 col-md-6">
        <label for="marque">Marque du véhicule</label>
        <select class="form-select" name="marque" id="">
            <option selected>
<?=$currentVehicule['marque_nom'] ?></option>
            <?php foreach ($marques as $marque) : ?>
                <option value="<?=$marque['marque_nom']; ?>">
                    <?=$marque['marque_nom']; ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

    <div class="col-lg-3 col-md-6">
        <label for="modele">Modele du véhicule</label>
        <select class="form-select" name="modele" id="">
            <option selected>
<?=$currentVehicule['modele_nom'] ?></option>
            <?php foreach ($modeles as $modele) : ?>
                <option value="<?=$modele['modele_nom']; ?>">
                    <?=$modele['modele_nom']; ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

    <div class="col-lg-3 col-md-6">
        <label for="type">Type du véhicule</label>
        <select class="form-select" name="type" id="">
            <option selected>
<?=$currentVehicule['type_nom'] ?></option>
            <?php foreach ($types as $type) : ?>
                <option value="<?=$type['type_nom']; ?>">
                    <?=$type['type_nom']; ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>
</div>
</div>

```

```

<div class="row pt-4 pb-3">

    <div class="col-lg-3 col-md-6">

        <div class="form-group <?=$immatriculationFalse
?>">

            <label for="plaqueImmatriculation">Plaque
d'Immatriculation</label>

                <input type="text"
name="plaqueImmatriculation" pattern="^[A-Z]{2}+[0-9]{3}+[A-Z]{2}+$"
                class="form-control" placeholder=""
value="<?=$currentVehicule['plaque'];?>">
            </div>

        </div>

        <?php if($_SESSION['fk_emploi'] == 4): ?>
        <div class="col-lg-3 col-md-6">

            <label for="InventairePosition">Inventaire
Position</label>

                <select class="form-select"
name="inventairePosition" id="">

                    <option selected>
<?=$inventairePosition['societe_nom'] ?></option>
                    <?php foreach ($societes as $societe) : ?>
                        <option
value="<?=$societe['societe_nom'];?>"><?=$societe['societe_nom'];?></option>
                    <?php endforeach; ?>
                </select>

            </div>

            <?php endif; ?>
            <div class="col-lg-3 col-md-6">
                <div class="form-group">
                    <!-- Date input -->
                    <label class="control-label"
for="datemiseecirculation">Date de mise en circulation
                    </label>
                    <input class="form-control" id="date"
name="datemiseecirculation" type="date"
                    value="<?=$currentVehicule['date_mise_circulation'];?>" />

```

```
</div>
</div>
        <?php if($_SESSION['fk_emploi'] == 4): ?>
<div class="col-lg-3 col-md-6">

        <label for="InventairePosition">Cession du
véhicule</label>

        <select class="form-select" name="cession" id="">
            <option selected value="En activité">En
activité </option>

            <option value="A céder">A céder </option>
        </select>

    </div>

<?php endif; ?>

<footer class="footer container pb-4 text-end mt-4">

    <div class="row ">

        <div class="col-md-12 mt-4">
            <div class="form-group ">
                <!-- Submit button -->
                <button class="btn btn-<?=$colorBtn?"
name="modifyGestionnaire"
                                type="submit">Je valide mes
modifications </button>

            </div>
        </div>

    </div>

</div>

</div>

</div>
```

```

        </form>
    </div>
<?php endif; ?>

<!-- CREATION VEHICULE -->

<?php if(isset($_POST['create'])): ?>

<!-- REMAQUE FORMULAIRE -->

<div class="card">
    <div class="card-header">
        <h1 class="text-= $colorBtn ?&gt;"&gt;Cr  er un V  hicule&lt;/h1&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;

        &lt;form method="POST" class="was-validated" action="&lt;?php
echo htmlspecialchars($_SERVER["PHP_SELF"]);?&gt;"&gt;

            &lt;div class="container "&gt;
                &lt;div class="row justify-content-between"&gt;

                    &lt;div class="border bg-whiteBlue p-3
col-md-5"&gt;

                        &lt;!-- FIRST CARD --&gt;
                        &lt;h2 class="display-3"&gt;

                            &lt;?php $userSoc =
getSocieteById($_SESSION['fk_societe']) ?&gt;
                            &lt;?= ($_SESSION['fk_emploi'] == 4) ?
$_SESSION['createCurrentSociete'] : $userSoc['societe_nom'] ?&gt;&lt;/h2&gt;
                            &lt;h4 class="mt-5"&gt;
                                &lt;?php if($_SESSION['fk_emploi'] == 4): ?&gt;
                                    &lt;?= $_SESSION['marqueVehicule'] ?&gt;
                                &lt;?php else: ?&gt;
                                    &lt;?=$_SESSION['marqueVehicule']; ?&gt;
                                &lt;?php endif; ?&gt;
                            &lt;/h4&gt;
</pre

```



```

<div class="mt-3 ">
    <label class="form-label"
for="modele">Modele du véhicule</label>
    <select class="form-select" required
aria-label="select example" id="validationTextarea" name="modele">
        <option value="">Choisir un
modèle</option>
        <?php foreach ($modeles as $modele) :
?>
            <option
value="<?=$modele['modele_nom']; ?>">
                <?=$modele['modele_nom']; ?>
            </option>
        <?php endforeach; ?>
    </select>
</div>

<div class="mt-3">
    <label for="type">Type du
véhicule</label>
    <select class="form-select" required
aria-label="select example" name="type" id="">
        <option value="">Choisir un
type</option>
        <?php foreach ($types as $type) : ?>
            <option
value="<?=$type['type_nom']; ?>">
                <?=$type['type_nom']; ?>
            </option>
        <?php endforeach; ?>
    </select>
</div>

<div class="mt-3">
    <label for="type">Alias du
véhicule</label>
    <input class="form-control
is-invalid"
id="alias"
aria-describedby="inputGroupPrepend3
validationServerUsernameFeedback"
required
type="text"

```

```

        name="alias"
        placeholder="Alias du véhicule">
    </div>

</div>

<!-- SECOND CARD -->
<div class="border p-2 col-md-6">

    <div class="mt-5">
        <label class="form-label"
for="plaqueImmatriculation">Plaque d'Immatriculation</label>
        <input type="text"
class="form-control is-invalid"
        id="validationImmatriculation"
        aria-describedby="inputGroupPrepend3
validationServerUsernameFeedback"
        required
        name="plaqueImmatriculation"
        pattern="[A-Z]{2}[0-9]{3}[A-Z]{2}"
        placeholder="immatriculation">
    </div>

    <div class="mt-3">
        <label
for="kilometrage">Kilometrage</label>
        <input class="form-control
is-invalid"
        id="validationImmatriculation"
        aria-describedby="inputGroupPrepend3
validationServerUsernameFeedback"
        required
        type="text"
        name="kilometrage"
        pattern="[0-9]*"
        placeholder="Kilometrage">
    </div>

    <div class="mt-3">
<?php if($_SESSION['fk_emploi'] == 4): ?>

```

```

</label>
for="InventairePosition">Inventaire Position</label>
                                <select class="form-select"
name="inventairePosition" id="">
                                <option selected>
<?=$_SESSION['createCurrentSociete']?></option>
                                <?php foreach ($societes as
$societe) : ?>
                                <option
value="<?=$societe['societe_nom'];?>"><?=$societe['societe_nom'];?>
                                </option>
                                <?php endforeach; ?>
                                </select>
<?php endif; ?>
                                </div>

                                <div class="mt-3">
                                <div class="form-group">
                                <!-- Date input -->
                                <label class="control-label"
for="datemisecirculation">Date de mise en circulation </label>
                                <input class="form-control" id="date"
name="datemisecirculation" type="date"
                                value="<?=$currentDate ?>" />
                                </div>
                                </div>

                                <div class="mt-4 ">
                                <div class="form-group">
                                <!-- Date input -->
                                <label class="control-label"
for="dateControleTechnique">Date de mise du controle technique </label>
                                <input class="form-control" id="date"
name="dateControleTechnique" type="date"
                                value="" />
                                </div>
                                </div>

                                <!-- Submit button -->
                                <div class="w-95 mt-3">

```

```

                                <button class="btn btn-<?=$colorBtn?> w-100 "
name="createGestionnaire" type="submit">Créer le
                                véhicule</button>
                                </div>

                                </div>
                                </div>
                                </div>
                                </form>
                                </div>
                                </div>

<?php endif; ?>

                                <script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
                                <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"><
/s>script>

</body>

</html>

```

Contenu de formulaireVueModel.php (Modifier et Créer un véhicule)

```

<?php
require ('../model/formulaireModel.php');
if ($_SESSION['fk_emploi'] == 3) {
    $redirectPath = 'reguleForm.php';
}
if ($_SESSION['fk_emploi'] == 4) {
    $redirectPath = 'gestionnaireForm.php';
}
/**
 * Function cleanStringForm
 * @param string $str
 * @param string $format
 * @return string $str
 */

```

```

function cleanStringForm($str){
    $str = htmlspecialchars($str);
    if(preg_match ('/^[A-Z]{2}+[0-9]{3}+[A-Z]{2}+$/ ', $str)) {
        return $str;
    }else{
        return false; }
}

/*****
/*****
/*****      MODIFIER UN VEHICULE EXISTANT      *****/
/*****
/*****

if (isset($_POST['modify'])) {
    if($_POST['alias']!='Selectionner votre véhicule'){
        $currentAlias = $_POST['alias'];
        $currentVehicule = getVehiculeData($currentAlias);
        $marques = nameFetch('marque');
        $modeles = nameFetch('modele');
        $societes = nameFetch('societe');
        $types = nameFetch('type');
        $_SESSION['alias'] = $currentVehicule['ancien_id'];
        $inventairePosition =
getSocieteById($currentVehicule['fk_inventaire_position']);
        $immatriculationFalse = "";
    }else{
        ($_SESSION['fk_emploi'] == 3) ? header('location:reguleForm.php') :
header('location:gestionnaireForm.php');
    }
}

if (isset($_POST['modifyGestionnaire'])) {
$plaqueImmatriculation = trim($_POST['plaqueImmatriculation']);
    if (cleanStringForm($plaqueImmatriculation) == false){
        $immatriculationFalse = " bg-danger ";
    }else{
$immatriculationFalse = "";
$alias = $_SESSION['alias'];
        if ($_SESSION['fk_emploi'] == 3){
            $societe =$_SESSION['fk_societe'];
        } else {
            $societeStr =trim( $_POST['societe']);
            $societe = (getSocieteByName($societeStr));
        }
$marqueStr = trim($_POST['marque']);

```

```

$marque = getMarqueByName($marqueStr);
$modeleStr = trim($_POST['modele']);
$modele = getModeleByName($modeleStr);
$typeStr = trim($_POST['type']);
$type = getTypeByName($typeStr);
    if ($_SESSION['fk_emploi'] == 3){
        $inventairePosition = getInventairePosition($alias);
    }else{
        $inventairePositionStr = trim($_POST['inventairePosition']);
        $inventairePosition = getSocieteByName($inventairePositionStr);
    }
$datemisecirculation = $_POST['datemisecirculation'];
$cessionStr = $_POST['cession'];
($cessionStr == "En activité") ? $cession = 0 : $cession = 1 ;
updateVehicule(
    $Emploi = $_SESSION['fk_emploi'],
    $alias,
    ($_SESSION['fk_emploi'] == 3) ? $societe : $societe['id_societe'],
    $marque['id_marque'],
    $modele['id_modele'],
    $type['id_type'],
    $plaqueImmatriculation,
    ($_SESSION['fk_emploi'] == 3)
);
$inventairePosition['fk_inventaire_position']
$inventairePosition['id_societe'],
    $datemisecirculation,
    $cession
);
}
}

/*****
/*****
/*****    CREER UN VEHICULE    *****/
/*****
/*****
/*****

if (isset($_POST['create'])) {
    //Si le profil est celui du gestionnaire
if ( $_SESSION['fk_emploi'] == 4) {
    if($_POST['marqueCreate']!='Selectionner une marque' &&
$_POST['societeCreate']!='Selectionner une société' ){
        $_SESSION['createCurrentSociete'] = $_POST['societeCreate'];
        $_SESSION['marqueVehicule'] = $_POST['marqueCreate'];
    }
}
}

```

```

        $modeles = getModelByMarque($_POST['marqueCreate']);
        $societes = nameFetch('societe');
        $types = nameFetch('type');
        $currentDate = date('Y-m-d');
    }else{
        ($_SESSION['fk_emploi'] == 3) ? header('location:reguleForm.php')
: header('location:gestionnaireForm.php');
    }
} else {
    $modeles = getModelByMarque($_POST['marqueCreate']);
    $types = nameFetch('type');
    $currentDate = date('Y-m-d');
    $_SESSION['marqueVehicule'] = $_POST['marqueCreate'];
}}
if(isset($_POST['createGestionnaire'])){
    if ($_SESSION['fk_emploi'] == 3){
        $societe = $_SESSION['fk_societe'];
        $marque = getModelByMarque($_SESSION['marqueVehicule']);
    }else{
        $societeStr= $_SESSION['createCurrentSociete'];
        $societe = getSocieteByName($societeStr);
        $marqueStr = trim($_SESSION['marqueVehicule']);
        $marque = getModelByMarque($marqueStr);
        // CHECK ALIAS IN BDD
        $alias = htmlspecialchars(trim($_POST['alias']));
        $fetchDoublon = getAliasDoublon($alias);
        $checkDoublon = ($fetchDoublon == $alias )?true:false;
        if ($checkDoublon == false) {
            echo "<div class='text-center my-5 py-3'";
            echo "<h2 class='mt-5 text-center mx-auto h3 '> ERREUR ALIAS
EXISTANT <h2>";
            echo "<br/>";
            echo "<a class='btn btn-secondary h3 mx-auto'
href=\".$redirectPath.\"> Retourner sur mon profil </a>";
            echo "</div>";
        }
        $modeleStr = trim($_POST['modele']);
        $modele = getModelByName($modeleStr);
        $typeStr = trim($_POST['type']);
        $type = getTypeByName($typeStr);
        $immatriculation= htmlspecialchars($_POST['plaqueImmatriculation']);
        if ($_SESSION['fk_emploi'] == 3){
            $inventaire = $societe;

```

```

    }else{
        $inventaire = trim($_POST['inventairePosition']);
        $inventaire = getSocieteByName($inventaire);
    }
    $dateMiseCirculation= $_POST['dateMiseCirculation'];
    $dateControleTechnique= $_POST['dateControleTechnique'];
    $kilometrage= htmlspecialchars($_POST['kilometrage']);
    if (strlen($dateControleTechnique) == 0 || strlen($dateControleTechnique)
> 10 ) {
        $dateControleTechnique = '2000-01-01';
    }
    if ($dateMiseCirculation == '') {
        $dateMiseCirculation = "2000-01-01";
    }
    $Emploi = $_SESSION['fk_emploi'];
    $validate = false;
    if ($_SESSION['fk_emploi'] == 3 && $checkDoublon != false){
        $validate = insertIntoVehicule(
            $Emploi,
            $societe,
            $marque['id_marque'],
            $modele['id_modele'],
            $type['id_type'],
            $immatriculation,
            $inventaire,
            $dateMiseCirculation,
            $dateControleTechnique,
            $kilometrage,
            $alias
        );
    }elseif ($_SESSION['fk_emploi'] == 4 && $checkDoublon != false){
        $validate = insertIntoVehicule(
            $Emploi,
            $societe['id_societe'],
            $marque['id_marque'],
            $modele['id_modele'],
            $type['id_type'],
            $immatriculation,
            $inventaire['id_societe'],
            $dateMiseCirculation,
            $dateControleTechnique,
            $kilometrage,
            $alias

```



```

    );
}

/** CREATION DU QR CODE : */
if ( $validate == true) {
    $_SESSION['newVehiculeImmat'] = $immatriculation;
    header('location: ../model/QrCodeConstruct.php');
}else{
    header('location: ../vue/erreur.php');
}
}
?>

```

Contenu de formulaireModel.php (créer et modifier un véhicule)

```

<?php
if ($_SESSION['fk_emploi'] == 3) {
    $redirectPath = 'reguleForm.php';
}
if ($_SESSION['fk_emploi'] == 4) {
    $redirectPath = 'gestionnaireForm.php';
}
/**
 * getVehiculeData function
 * Retourne toutes les data concernant un véhicule par rapport à son alias.
 * @param [string] $alias
 * @return iterable
 */
function getVehiculeData($alias){
    $bd = connectBd();
    $sql = $bd->prepare("SELECT * from sanios_vehicule
    JOIN sanios_societe on fk_societe = id_societe
    JOIN sanios_marque on fk_marque = id_marque
    JOIN sanios_modele on fk_modele = id_modele
    JOIN sanios_type on fk_type = id_type
    where ancien_id = :alias");
    $sql->bindParam('alias',$alias, PDO::PARAM_STR);
    $sql -> execute();
    $retourSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retourSql;
}
function getInventairePosition($alias){
    $bd = connectBd();

```

```

        $sql = $bd->prepare("SELECT fk_inventaire_position FROM sanios_vehicule
WHERE ancien_id = :alias");
        $sql->bindParam('alias', $alias);
        $sql->execute();
        $returnSql = $sql->fetch(PDO::FETCH_ASSOC);
        return $returnSql;
    }

/**
 * getSocieteById function
 * Récupère toutes les informations d'une société par son id
 * @param [int] $idSociete
 * @return array
 */
function getSocieteById($idSociete){
    $bd = connectBd();
    $sql = $bd->prepare("SELECT * FROM sanios_societe WHERE id_societe =
:idSociete ");
    $sql->bindParam('idSociete', $idSociete, PDO::PARAM_INT);
    $sql->execute();
    $retournSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retournSql;
}

/**
 * getSocieteByName function
 * Retourne l'id de la société par son nom
 * @param [string] $NomSociete
 * @return int
 */
function getSocieteByName($NomSociete){
    $bd = connectBd();
    $sql = $bd->prepare("SELECT `id_societe` from sanios_societe where
societe_nom = :NomSociete ");
    $sql->bindParam('NomSociete', $NomSociete, PDO::PARAM_STR);
    $sql->execute();
    $retournSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retournSql;
}

/**
 * getMarqueByName function
 * Récupère l'id d'une marque par rapport à son nom
 * @param [string] $NomMarque
 * @return int
 */

```

```

function getMarqueByName($NomMarque){
    $bd = connectBd();
    $sql = $bd->prepare("SELECT id_marque FROM sanios_marque WHERE marque_nom
= :NomMarque ");
    $sql->bindParam('NomMarque', $NomMarque, PDO::PARAM_STR);
    $sql->execute();
    $retournSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retournSql;
}

/**
 * getModeleByName function
 * Récupère l'id d'un modele par rapport à son nom
 * @param [string] $NomModele
 * @return int
 */
function getModeleByName($NomModele){
    $bd = connectBd();
    $sql = $bd->prepare("SELECT id_modele FROM sanios_modele WHERE modele_nom
= :NomModele ");
    $sql->bindParam('NomModele', $NomModele, PDO::PARAM_STR);
    $sql->execute();
    $retournSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retournSql;
}

/**
 * getTypeByName function
 * Récupère l'id d'un type par rapport à son nom
 * @param [string] $NomType
 * @return int
 */
function getTypeByName($NomType){
    $bd = connectBd();
    $sql = $bd->prepare("SELECT id_type FROM sanios_type WHERE type_nom =
:NomType ");
    $sql->bindParam('NomType', $NomType, PDO::PARAM_STR);
    $sql->execute();
    $retournSql = $sql->fetch(PDO::FETCH_ASSOC);
    return $retournSql;
}

/**
 * getModelByMarque function
 * Récupère les modèles affiliés à une marque
 * @param [string] $nomMarque

```

```

* @return array
*/
function getModelByMarque($nomMarque) {
    $bd = connectBd();
    $sql= $bd->prepare("SELECT modele_nom FROM `sanios_modele` join
sanios_marque on fk_marque = id_marque where marque_nom = :nomMarque");
    $sql->bindParam('nomMarque', $nomMarque, PDO::PARAM_STR);
    $sql->execute();
    $retourSql = $sql->fetchAll(PDO::FETCH_ASSOC);
    return $retourSql;
}

/**
 * getAliasDoublon function
 * Vérifie les doublons d'ancien_id
 * @param [string] $alias
 * @return boolean
 */
function getAliasDoublon($alias){
    $bd = connectBd();
    $sql=$bd->prepare('SELECT ancien_id from sanios_vehicule where ancien_id
= :alias ');
    $sql->bindParam('alias',$alias,PDO::PARAM_STR);
    $sql->execute();
    $retournSql = $sql->fetch(PDO::FETCH_ASSOC);
    if ($retournSql == false ) {
        return $alias;
    }else{
        return $retournSql;
    }
}

/**
 * updateVehicule function
 * Modification d'un véhicule
 * @param [string] $alias
 * @param [int] $societe
 * @param [int] $marque
 * @param [int] $modele
 * @param [int] $type
 * @param [string] $plaqueImmatriculation
 * @param [string] $inventairePosition
 * @param [string] $datemisecirculation
 * @param [bool] $cession
 * @return header

```

```

*/
function updateVehicule($Emploi, $alias, $societe, $marque, $modele, $type,
$plaqueImmatriculation, $inventairePosition, $datemisecirculation, $cession){
    $bd = connectBd();
    $sql = $bd->prepare("UPDATE `sanios_vehicule`
        SET `plaque` = :immatriculation , `fk_marque` = :idMarque, `fk_modele` =
:idModele, `fk_type` = :idType, `fk_societe` = :idSociete,
        `fk_inventaire_position` = :inventairePosition, `date_mise_circulation` =
:dateMiseCirculation, `cession` = :cession
        WHERE `ancien_id` = :alias");
    $sql->bindParam('alias',$alias);
    $sql->bindParam('idSociete',$societe);
    $sql->bindParam('immatriculation',$plaqueImmatriculation);
    $sql->bindParam('idMarque',$marque);
    $sql->bindParam('idModele',$modele);
    $sql->bindParam('idType',$type);
    $sql->bindParam('inventairePosition',$inventairePosition);
    $sql->bindParam('dateMiseCirculation',$datemisecirculation);
    $sql->bindParam('cession',$cession);
    $sql ->execute();
    return ($Emploi == 3) ? header('location:reguleForm.php') :
header('location:gestionnaireForm.php');
}

/**
 * insertIntoVehicule function
 * Insertion dans la bdd d'un véhicule
 * @param [int] $societe
 * @param [int] $marque
 * @param [int] $modele
 * @param [int] $type
 * @param [string] $immatriculation
 * @param [int] $inventaire
 * @param [string] $dateMiseCirculation
 * @param [string] $dateControleTechnique
 * @param [int] $kilometrage
 * @param [string] $alias
 * @return header
 */
function insertIntoVehicule($Emploi,$societe, $marque, $modele,
$type,$immatriculation,$inventaire,$dateMiseCirculation,$dateControleTechniqu
e, $kilometrage,$alias){
    $bd = connectBd();
    $sql = $bd->prepare("INSERT INTO `sanios_vehicule`

```

```

        (`id_vehicule`,`plaque`,`fk_marque`,`fk_modele`,`fk_type`,
`fk_societe`,`fk_inventaire_position`,`date_mise_circulation`,
`date_controle_technique`,`kilometrage`,`cession`,`ancien_id`)
        VALUES (NULL, :immatriculation, :marque, :modele, :typev, :societe,
:inventaire, :dateCirculation, :dateControleTechnique, :kilometrage, '0',
:alias);");
    $sql->bindParam('societe',$societe);
    $sql->bindParam('marque',$marque);
    $sql->bindParam('modele',$modele);
    $sql->bindParam('typev',$type);
    $sql->bindParam('immatriculation',$immatriculation);
    $sql->bindParam('inventaire',$inventaire);
    $sql->bindParam('dateCirculation',$dateMiseCirculation);
    $sql->bindParam('dateControleTechnique',$dateControleTechnique);
    $sql->bindParam('kilometrage',$kilometrage);
    $sql->bindParam('alias', $alias);
    $validate = ($sql ->execute()) ? true : false ;
    return $validate;
}
?>

```