Rapport projet SGBD:
Entretien de véhicules (Garage)

Groupe 4

Membres : Théophane Loloum, Quentin Chavigny, Sam Gubernator, Arthur Hermitte

Résumé : L'objectif du projet est de mettre en œuvre, sur un cas pratique, les notions et les méthodes vues dans le module SGBD. Le projet démarre par une modélisation des données, et aboutit à la création d'une base de données relationnelle et à l'implémentation d'un certain nombre d'opérations (consultations, mises à jour, etc.).

1 Modélisation des données

1.1 Description de la base

L'objectif de cette base de données est de simuler un garage interagissant avec des clients, leurs véhicules, et de noter des informations relatives aux interventions ayant lieu sur ces véhicules.

Pour mettre en place une base de données cohérente avec ces objectifs, il a fallu tout d'abord définir des entités :

1. Clients
2. Vehicules
3. Types_vehicule
4. Pieces
5. Types_piece
6. Interventions
7. Prises_en_charge
8. Garages

Au cours de la conception, cette liste a évolué pour atteindre cet état. Par exemple, l'entité "garage" n'a été ajoutée qu'après que le problème ne se soit posé de vouloir noter un historique d'interventions dans des garages externes. De même, le parti-pris de séparer les notions d'intervention et de prise en charge découle de l'interprétation personnelle par notre groupe suivante :

Un client va déposer un ou plusieurs de ses véhicules dans notre garage, ce qui donnera lieu à la création d'une prise en charge. Au cours de cette prise en charge, chacun de ses véhicules subira zéro, une ou plusieurs interventions individuelles.

Ainsi, le prix payé par le client à la fin de la prise en charge se retrouve être la somme des prix de main d'œuvre liés aux interventions qui ont eu lieu et des prix des pièces qui ont dû être utilisées pendant celles-ci. Ce calcul est relativement complexe, mais cela aurait induit de la duplication d'information de créer un champ 'prix_prise_en_charge'.

De même, pour informer d'une intervention prévue pour une date future, plutôt que de créer une entité 'Prescriptions' qui ferait doublon, on préférera simplement entrer une intervention avec une date de début future, en laissant vide les champs dont l'information est pour l'instant manquante, comme la durée ou les pièces utilisées.

Pour faire référence à une intervention qui a eu lieu dans un autre garage, il n'est pas nécessaire d'avoir non plus de prise en charge associée, ni d'entrer le tarif horaire du garage (seules les informations techniques sont intéressantes, le garage ne s'intéresse pas aux prix payés par les clients pour les interventions externes).

Les véhicules possèdent tous un unique type qui leur est associé, tout comme les pièces possèdent aussi un unique type qui leur est associé. Il faut comprendre cela comme la différence entre un objet ayant une réalité physique et le concept de cet objet.

Les cardinalités sont également majoritairement des partis pris : une pièce n'a lieu d'être dans la base qu'à partir du moment où celle-ci a été utilisée au cours d'une unique intervention, de même pour un garage ou un type de véhicule, dont l'information n'importe que si au minimum une intervention ou un véhicule y font référence.

De même, il doit y avoir des contraintes classiques sur les attributs de dates pour rester en cohérence : une intervention ne peut pas avoir lieu sur un véhicule antérieurement à la date de mise en circulation de celui-ci. La fin d'une prise en charge doit être postérieure au début de celle-ci...

1.2 Modèle entité-association

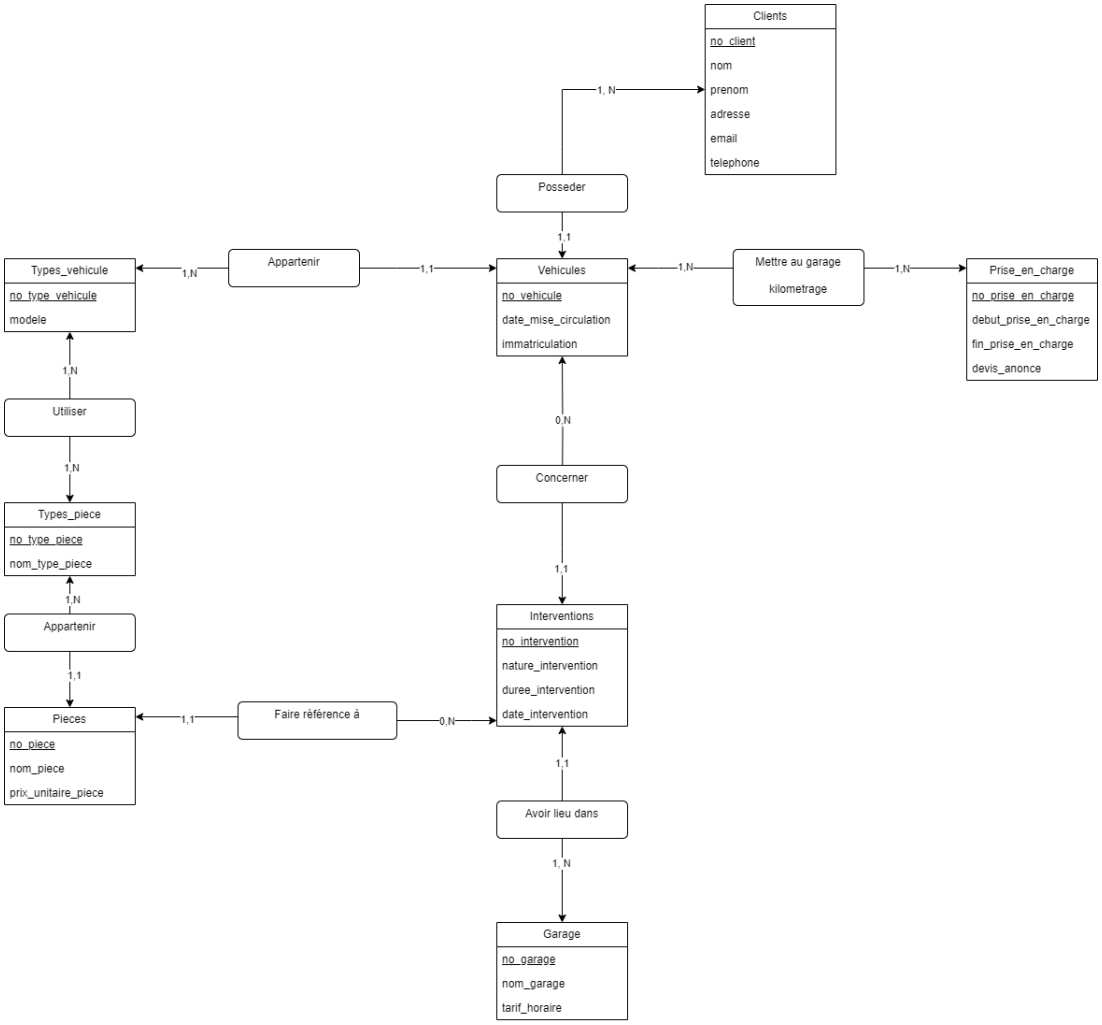

FIGURE 1 – Modèle conceptuel de la base

Soit le modèle entité-association présenté en Figure 1. Remarquons que deux associations ('Utiliser' et 'Mettre_au_garage') possèdent des cardinalités du type N :M et donneront donc lieu à la création de tables dédiées. Remarquons également l'existence d'un cycle. Cependant, celui-ci reste en cohérence avec l'objectif de normalisation car les données n'entrent pas en contradiction. Le cycle existe pour décrire les types de pièces et de véhicules, n'apportant pas de confusion particulière.

1.3 Opérations prévues

Vu la démultiplication des opérations possibles avec le développement de la base, toutes les possibilités ne seront pas traitées. Voici donc une liste non-exhaustive des opérations implémentées :

1. Consultations :
   (a) Les modèles de véhicules pour une année donnée
   (b) Les interventions réalisées pendant un intervalle donné
   (c) Liste des interventions prévues dans les deux mois.

(d) Liste des clients avec le nombre de véhicules qu'ils ont confiés au garage.
(e) Liste des modèles de véhicule pris en charge lors de l'année écoulée.

2. Statistiques :
   (a) La liste des clients, avec le total des sommes facturées à chacun.
   (b) Le nombre d'heures facturées par mois.
   (c) La liste des types de véhicules, avec le type d'intervention majoritaire pratiqué sur ces véhicules.

2 Passons au relationnel

À présent, le diagramme entité-association a été bien réfléchi, dans l'objectif d'une implémentation cohérente en SQL a posteriori. Le passage au relationnel sera donc relativement aisé. Comme dit préalablement, les associations N :M deviennent des entités dont la clé primaire est un tuple de clés étrangères référençant les clés primaires des entités en relation.

2.1 Contraintes d'intégrité

L'ajout dans la base de données de contraintes d'intégrités permet de s'assurer d'une certaine cohérence des données. On ajoutera notamment des contraintes de non nullité. Parmi les contraintes on retrouvera par exemple :

1. La cohérence des dates de début et de fin de prise en charge
2. La vérification que le tarif horaire d'un garage est renseigné si et seulement si il s'agit de notre garage
3. La vérification que les dates d'intervention sur un véhicule sont postérieures à sa mise en circulation
4. La vérification que le kilométrage renseigné lors d'une mise au garage ne fait qu'augmenter
5. La vérification qu'un client possède un véhicule associé et qu'un véhicule possède une intervention associée

2.2 Schéma relationnel

Il vient de ces considérations le schéma relationnel suivant :

Listing 1 – Schéma relationnel

```
Clients(no_client, <--- clef(s) primaire(s)
    nom, prenom, adresse, email, telephone) <--- autre(s) attribut(s)

Type_piece(no_type_piece,
    nom_type_piece)

Type_vehicule(no_type_vehicule,
    modele)

Utiliser(#no_type_piece, #no_type_vehicule
    )

Vehicule(no_vehicule,
    date_mise_circulation, immatriculation, #no_type_vehicule, #no_client)

Prise_en_charge(no_prise_en_charge,
    debut_prise_en_charge, fin_prise_en_charge, devis_annonce)

Garage(no_garage,
    nom_garage, tarif_horaire)
```

```
Mettre_au_garage(#no_vehicule, #no_prise_en_charge,
    kilometrage)

Intervention(no_intervention,
    nature_intervention, date_intervention, duree_intervention, #no_garage, #no_vehicule)

Piece(no_piece,
    nom_piece, prix_unitaire_piece, #no_type_piece, #no_intervention)
```

Pour chaque entité, le(s) attribut(s) en première ligne forme(nt) la clé primaire de l'entité.
Le schéma est bien en troisième forme normale, car :

1. Chaque attribut ne prend à un instant donné qu'une unique valeur.
2. Les attributs ne faisant pas partis de la clé primaire d'une entité dépendent tous de la clé primaire en entier.

2.3 Dépendances fonctionnelles

Il en découle le schéma des dépendances fonctionnelles suivant, basé sur les clés étrangères :

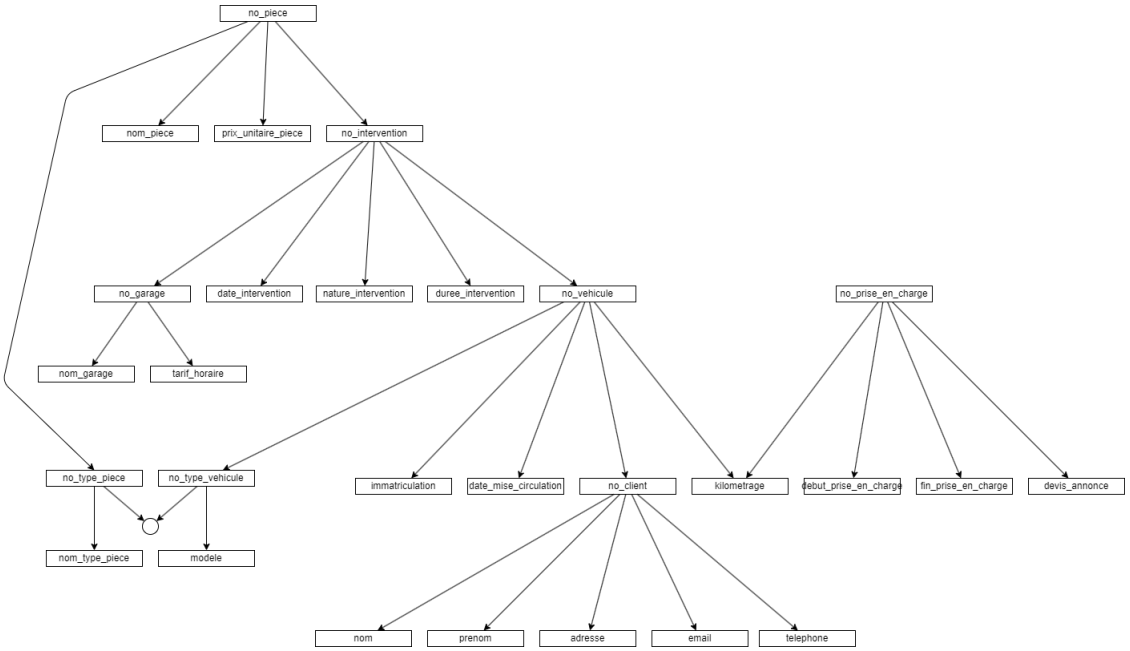

FIGURE 2 – Dépendances fonctionnelles

Par exemple `(no_vehicule, no_prise_en_charge)->kilometrage` et `no_piece->prix_unitaire_piece` (cf. Figure 2). Remarquons le cas particulier `(no_type_piece, no_type_vehicule)->0` issu de l'entité `Utiliser(#no_type_piece, #no_type_vehicule)`.

3 Implémentation

Lors de l'implémentation, pour rester dans la continuité des TDs, nous avons choisi d'utiliser une base PostgreSQL. La base de données est créée par le contenu du fichier create.sql, en suivant le modèle relationnel. On ajoute à cela des contraintes assurant la cohérence entre les différentes données de type date dans les différentes tables, comme expliquées ci-dessus, ou bien la cohérence des informations sur le garage (accès au tarif horaire dans notre garage uniquement). Concernant les données, on ajoute en dur des données avec le fichier insert.sql. A noter que dans la suite, lorsque le type de jointure n'est pas précisé, il s'agit d'un simple JOIN.

3.1 Requêtes

Cette partie traite des requêtes implémentées. Celles-ci sont retrouvables dans le fichier select.sql.

3.1.1 Consultation

On décrit ici les requêtes de consultation.

La requête fournissant les interventions prévues dans les 2 mois est effectuée par une simple sélection avec condition dans la table `interventions`.

La requête fournissant les clients et leur nombre de véhicules confiés au garage est effectuée en joignant les tables `client`,`vehicules`, puis en groupant sur les clients.

Celle fournissant la liste des modèles pris en charge dans l'année est effectuée en sélectionnant les noms des modèles dans la table `vehicules`, que l'on joint à la table `type_vehicule`, et en vérifiant que la date de début d'intervention correspond bien à l'année courante. On a interprété "année courante" par la date du jour, il y a un an (les 365 derniers jours et pas l'année 2022).

3.1.2 Statistiques

On décrit ici les requêtes fournissant des statistiques.

La requête fournissant la liste des clients ainsi que la somme totale qui leur a été facturée au cours d'une ou plusieurs interventions est effectuée en sélectionnant les clients auxquels on joint les véhicules, les interventions et les garages. On effectue également une jointure aux prises en charge pour s'assurer que l'on comptabilise bien uniquement les interventions de notre garage.

Concernant la requête renvoyant le nombre d'heures facturées par mois, on sélectionne le mois et l'année de la date d'intervention dans `interventions` ainsi que la somme des heures facturées. On groupe ensuite sur le mois et l'année.

Enfin, pour les interventions majoritaires par type de véhicule, on sélectionne les types de véhicules, le nombre total de nature d'interventions dans la table du même nom, ainsi que la nature d'intervention la plus courante en utilisant MAX(). On groupe ensuite selon les modèles.

3.2 Ajout de client

Nous avons également ajouté la possibilité d'ajouter des clients depuis la page d'accueil de la page web. À noter qu'on ne peut ajouter que des nouveaux clients, avec donc un nouveau véhicule, une nouvelle intervention, une prise en charge, etc. Toutes ces données sont ajoutées pour garder une cohérence avec le modèle (exemple : un client a forcément un véhicule sur lequel notre garage est intervenu).

4 Interface

Les résultats des requêtes peuvent être accédés via une page web.

4.1 Description

Arrivé sur la page, l'utilisateur est confronté à un menu (cf. Figure 3) lui permettant de naviguer vers une des trois pages suivantes : la page clients, la page consultations et la page statistiques.

— Clients (cf. Figure 4) : La page client permet de naviguer parmis l'ensemble des clients du garage et d'afficher des informations les concernanttelles que leur identité, leur adresse, leurs informations de contact, le nombre de véhicules confiés au garage par le client et le total des sommes facturées.

— Consultations (cf. Figure 5 et Figure 6) : La page consultation permet de visualiser la liste des modèles sur lequel le garage à intervenu, par an, ainsi que la liste des interventions passées ou prévues dans un interval de temps choisi.

— Statistiques (cf. Figure 7) : La page statistiques permet de se renseigner sur l'intervention la plus courante pour un certain type de véhicule ainsi que le total facturé sur une période choisie.

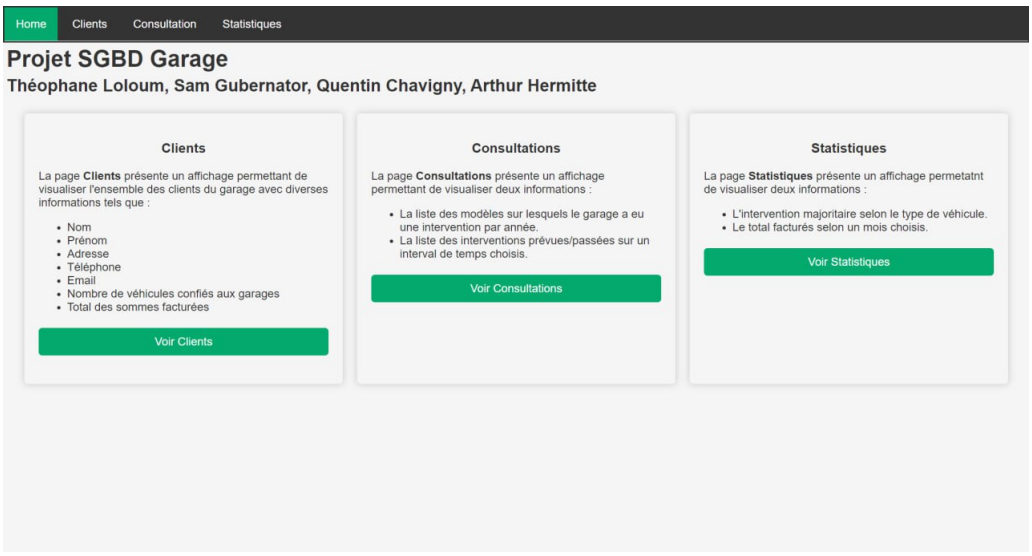

FIGURE 3 – Page d'accueil de l'interface web

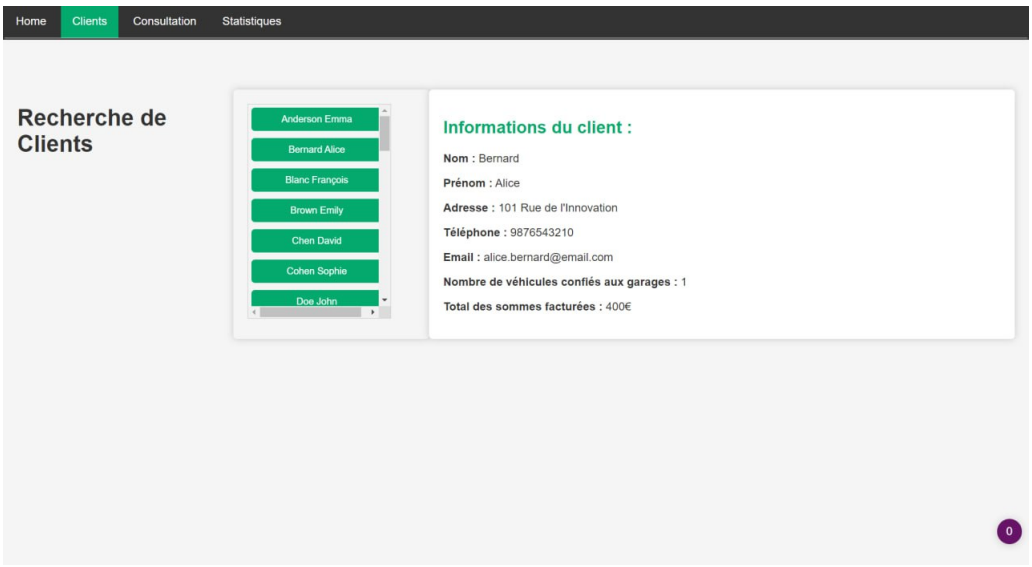

FIGURE 4 – Page clients de l'interface web

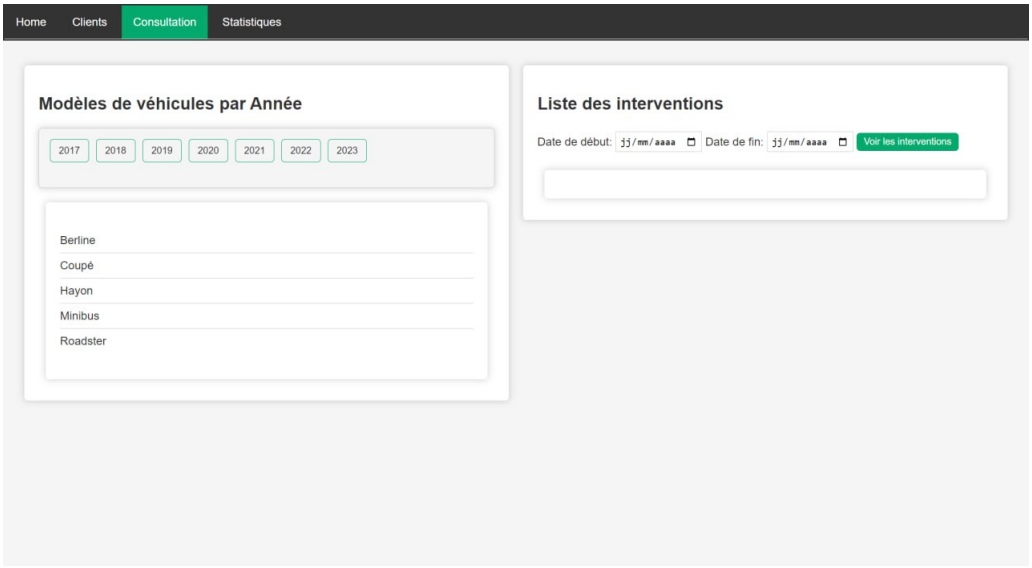

FIGURE 5 – Page de consultation de l'interface web (1)

4.2 Interactions avec la base de données

La page web est hébergée sur le serveur tloloum.zzz.bordeaux-inp.fr, et contient du code PHP permettant d'intéragir avec notre base de données. Nous avons choisi d'utiliser le language PHP d'une part afin de rester dans la continuité des TDs, mais également car il s'adaptait particulièrement à nos besoins tout en étant relativement simple d'utilisation et compatible avec une base PostgreSQL.

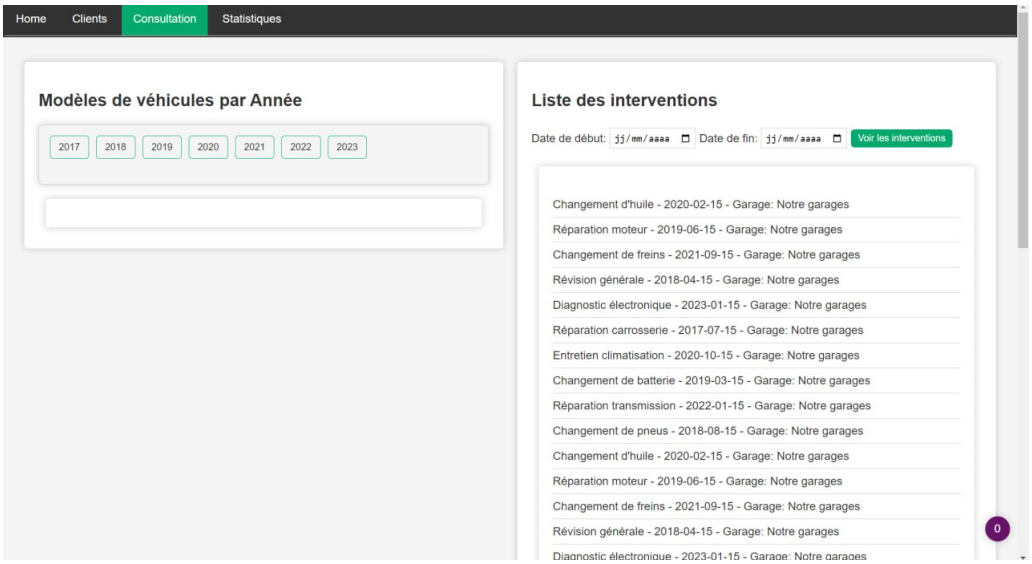

FIGURE 6 – Page de consultation de l'interface web (2)

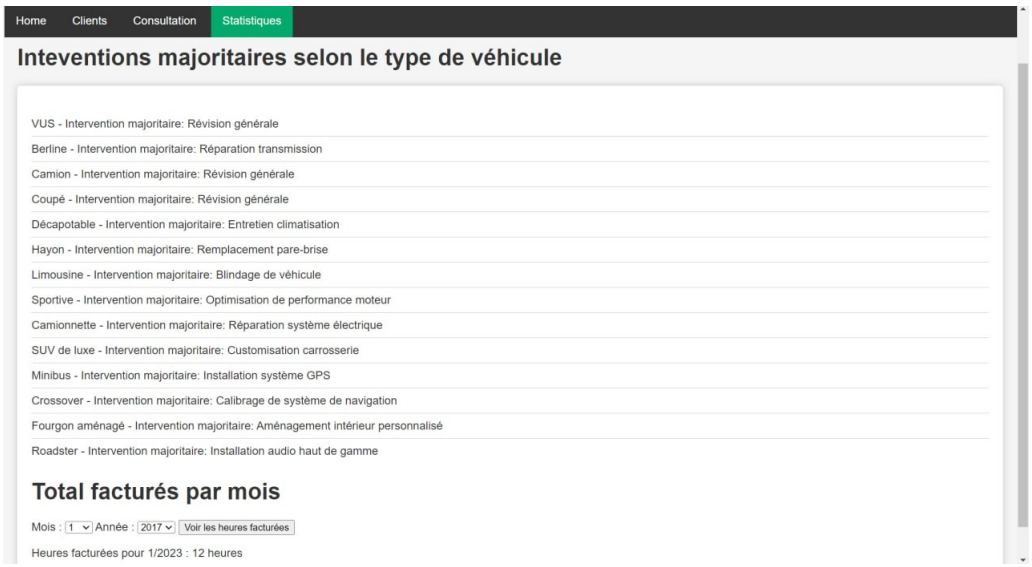

FIGURE 7 – Page des statistiques de l'interface web