

**E-GARAGE  
DOCUMENTATION**

Par  
Rakotonirainy Daniel Valisoa  
et  
Rakotondramaka Asandratra Mitia Ny Aina

Documentation pour  
le projet final du cours de Web Avancé M1 P12

Avril, 2025

Table des matières

[1. Introduction 3](#_Toc194581084)

[2. Accessibilité 4](#_Toc194581085)

[2.1. Compatibilité 4](#_Toc194581086)

[2.2. Procédure d’utilisation 4](#_Toc194581087)

[2.3. Sources 6](#_Toc194581088)

[2.3.1. DBGarageInit 6](#_Toc194581089)

[2.3.2. Backend 6](#_Toc194581090)

[2.3.3. Docs 6](#_Toc194581091)

[2.3.4. Frontend 6](#_Toc194581092)

[2.3.5. Stylized\_frontend 6](#_Toc194581093)

[3. Navigation et Interfaces 7](#_Toc194581094)

[3.1. Client 7](#_Toc194581095)

[3.2. Manager 8](#_Toc194581096)

[3.3. Mécanicien 8](#_Toc194581097)

[4. Fonctionnalités principales 8](#_Toc194581098)

[4.1. Prise de rendez-vous 8](#_Toc194581099)

[4.1.1. Demande de rendez-vous 9](#_Toc194581100)

[4.1.2. Rendez-vous 12](#_Toc194581101)

[4.2. Suivi de travail 13](#_Toc194581102)

[4.3. Suivi de paiement 13](#_Toc194581103)

[4.4. Enregistrement de congé 13](#_Toc194581104)

# Introduction

Pour le projet final du cours de Web Avancé, notre équipe, composée par Rakotonirainy Daniel et Rakotondramaka Asandratra, a mis en place une application web destinée à une chaîne de garages pour l’aider à assister sa clientèle.

L’application web a été pensée pour l’utilisation des employés du garage et surtout pour leurs clients.

Pour développer cette application, nous avons utilisé MEAN :

* MongoDB pour la base de données
* Express.js pour les services
* Angular pour l’interface utilisateur
* Node.js pour l’accessibilité aux services et fonctionnalités

Ce document a été rédigé pour fournir des détails sur l’application et sur son utilisation.

Il comporte les informations suivantes :

* Comment accéder et utiliser l’application web en question
* Les interfaces et comment naviguer de page en page sur l’application
* Les fonctionnalités principales que nous avons mis en place et leurs fonctionnements

# Accessibilité

L’application web est accessible sur l’adresse suivante :

<https://asandratra.github.io/m1p12mean-Daniel-Asandratra/>

Cette adresse accède à la version déployée avec GitHub Pages de l’interface utilisateur. L’interface fait appel aux services déployés sur Render.com, et les services utilisent une base de données en ligne sur MongoDB Atlas.

Il est à noter que due à l’utilisation de l’option gratuite de déploiement de Render.com, les services sont désactivés après une période d’inactivité. Si le service est inactif, il faudra attendre un certain temps pour son redémarrage lors de l’appel d’une fonction.

## Compatibilité

L’application a été testée sur

* Chrome, version 134.0.6998.178, dernière version du Mars 2025.

Les versions des outils utilisés pour l’application correspondent aux dernières versions supportées par les serveurs de déploiement datant du Mars 2025

* MongoDB 8
* Express.js 4
* Angular 19
* Node.js 22

## Procédure d’utilisation

Pour utiliser l’application, il faut d’abord avoir un profile enregistré dans la base de données.

Si vous n’avez pas encore de profile, il faut s’inscrire avec un pseudo unique, nom et prénom, le numéro de contact, mot de passe, et si vous vous inscrivez comme employé, dans la version actuelle de l’application, vous devez établir votre rôle, entre Manager ou Mécanicien, et le garage où vous allez travailler.

|  |  |
| --- | --- |
|  |  |

Après la création d’un profil, vous pouvez aller à la page d’authentification correspondant à votre profil, Client ou Employé, et vous connectez en entrant le pseudo et le mot de passe correctes pour accéder aux pages et aux fonctionnalités de l’application.

|  |  |
| --- | --- |
|  |  |

## Sources

Les codes sources de l’application sont sur GitHub dans un répertoire public à l’adresse suivante:

<https://github.com/Asandratra/m1p12mean-Daniel-Asandratra.git>

Les documents dans ce répertoire sont structurés comme suit

### DBGarageInit

Ce dossier contient des fichiers de données en format JSON pour initialiser la base en important les données directement.

Il contient également des notes de données pour voir certaines données crypter, tel que les mots de passe utilisé pour tester l’application.

### Backend

Ce document contient les services de l’application, avec les modèles de données, et les API pour faire appel aux fonctionnalités.

### Docs

Ce document résulte de la commande ‘ng build’ avec les sources de l’affichage de l’application, et en spécifiant le dossier de sortie comme ‘docs’.

C’est le document avec lequel GitHub Pages déploie l’interface utilisateur en ligne en format de SPA ou Single Page Application.

### Frontend

Ce dossier a été utiliser pour tester les fonctionnalités dans une interface non styliser.

Ce dossier a servi pour le développement et les tests locaux sans toucher aux interfaces principales.

### Stylized\_frontend

Ce dossier contient l’affichage plus soigné de l’application.

Cet affichage a été adaptée depuis un modèle fourni par Mantis-Free-Angular-Admin-Template

Dans ce dossier, les pages pour notre application se trouve dans ‘src/app/garage/’. Avec les pages pour les clients, les managers, et les mécaniciens respectivement dans les dossiers ‘client/’, ‘manager/’, ‘mecanic/’.

Pour modifier les routages, modifiez le fichier ‘src/app/app-routing.module.js’

# Navigation et Interfaces

En suivant l’adresse de l’application web, l’utilisateur sera dirigé vers la page d’authentification de client.

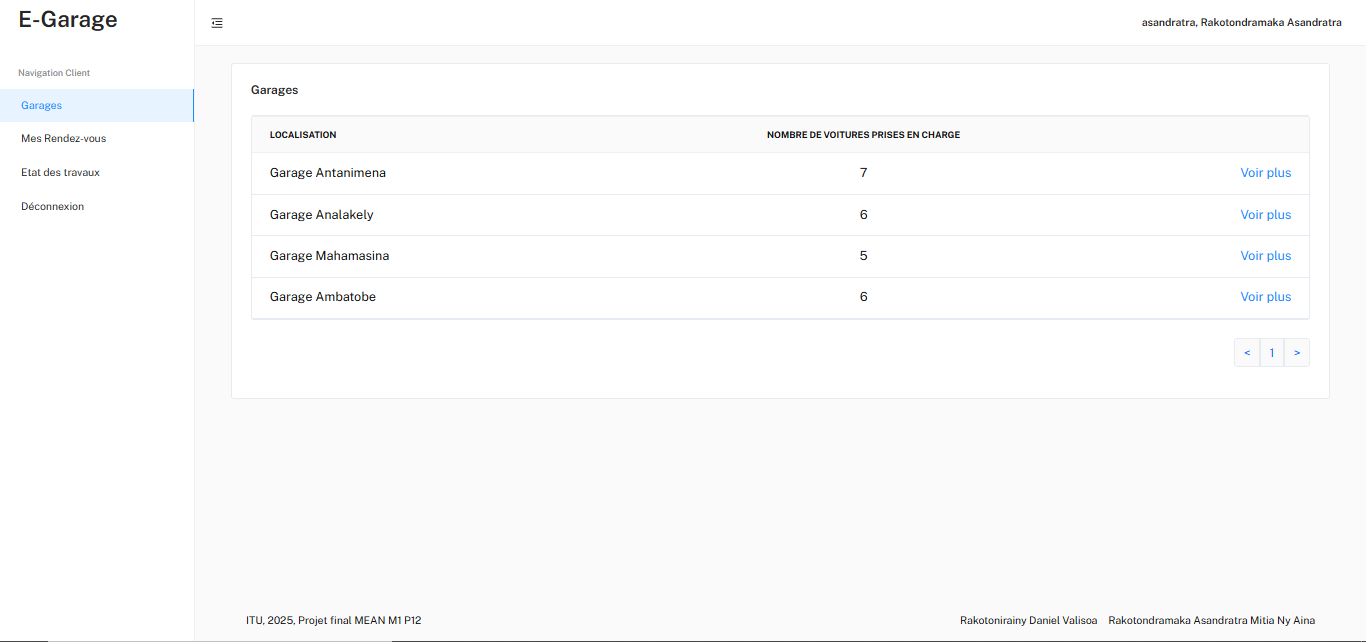
Dans la version actuelle de l’application, l’utilisateur peut passer à la page d’authentification des mécaniciens en cliquant sur un lien en dessous du formulaire d’authentification et inversement sur la page pour mécanicien.

Après authentification, il y a trois types d’interface pour cette application, une pour chaque profil utilisateur, client, manager, mécanicien. Chaque profil ayant un affichage similaire, une barre de navigation sur la gauche, mais des options différentes sauf pour la déconnexion pour revenir à la page d’authentification.

## Client

Un client sera dirigé vers la liste des garages après authentification.

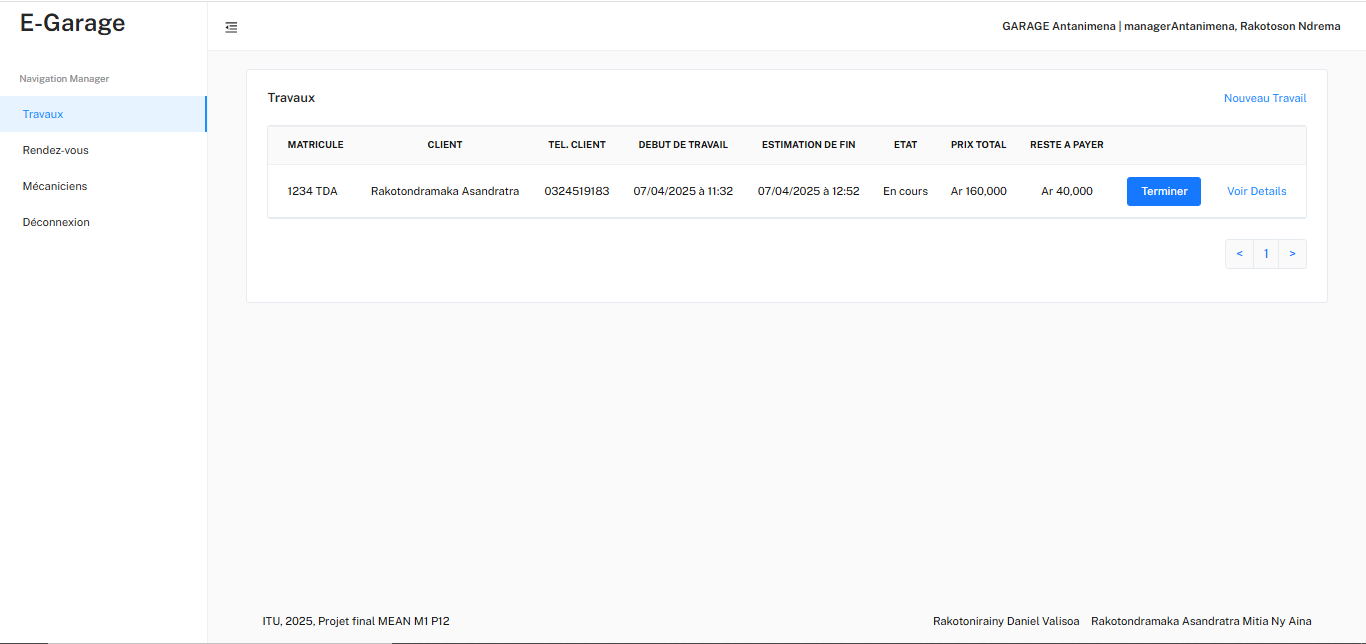
En utilisant la barre de navigation, un client peut naviguer parmi ces pages :

* Liste des garages
* Liste des rendez-vous du client
* Liste des travaux effectuer par le garage pour le client

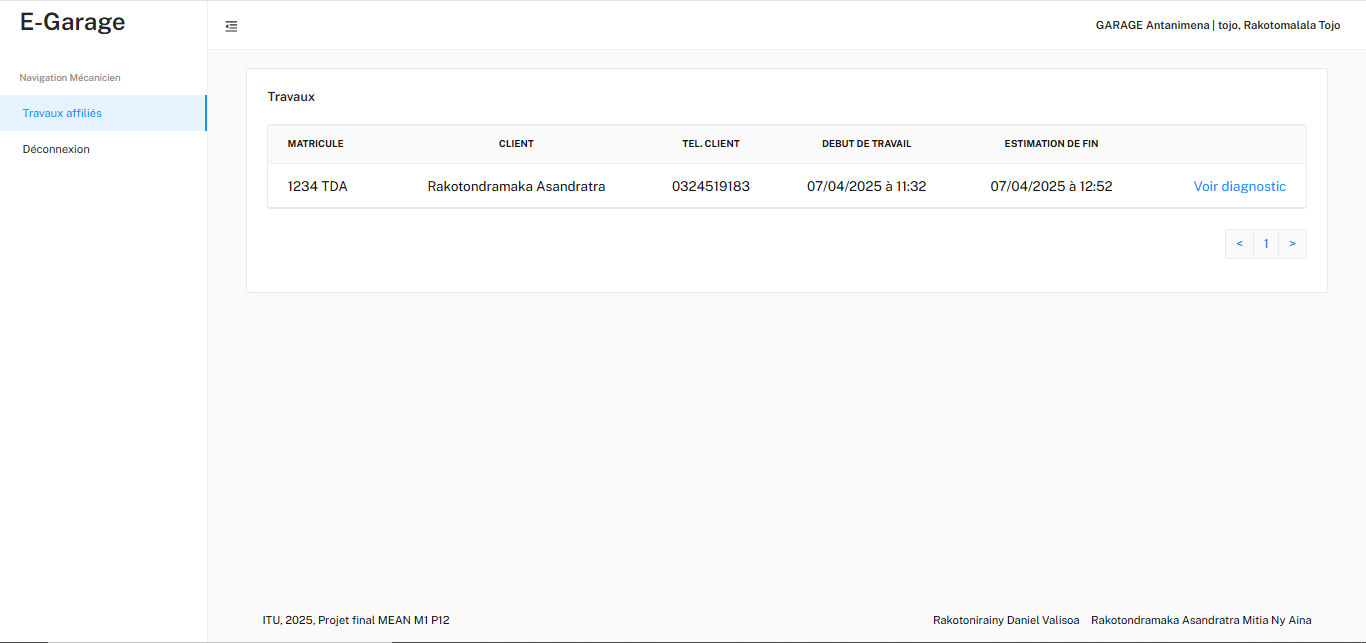
## Manager

Après authentification, un manager verra la page listant les travaux en cours du garage auquel il est affilié.

Le manager peut naviguer entre les pages suivantes :

* Liste des travaux du garage
* Liste des rendez-vous
* Liste des employés (manager et mécanicien)

## Mécanicien

Dans la version actuelle, l’option de navigation d’un mécanicien est la liste des travaux affiliés au mécanicien.

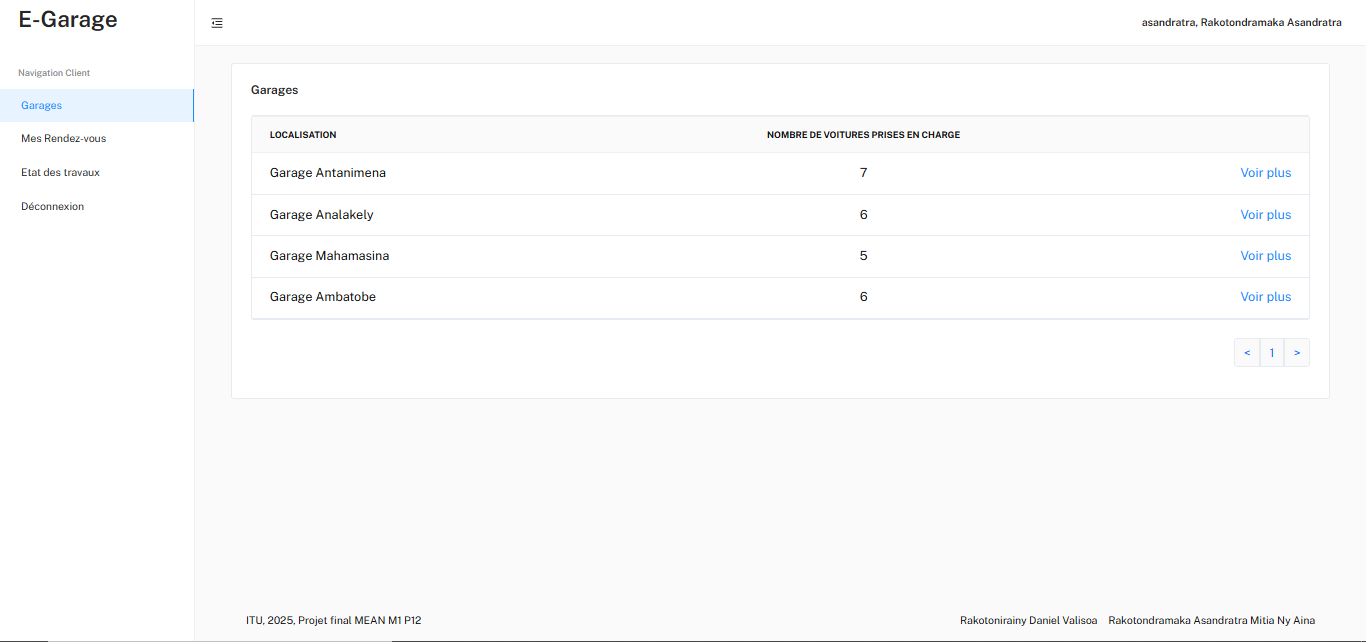
# Fonctionnalités principales

L’application web vise à l’utilisation des clients, et à servir aux garagistes. Nous avons alors mis en place les fonctionnalités suivantes :

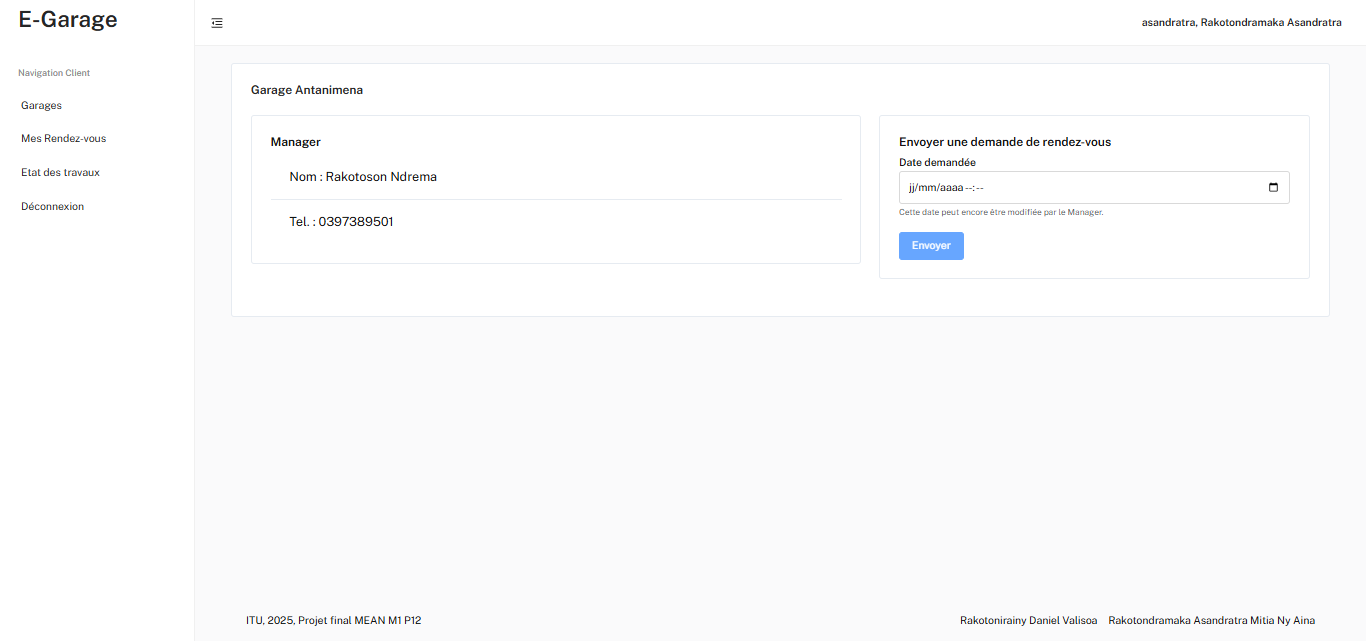
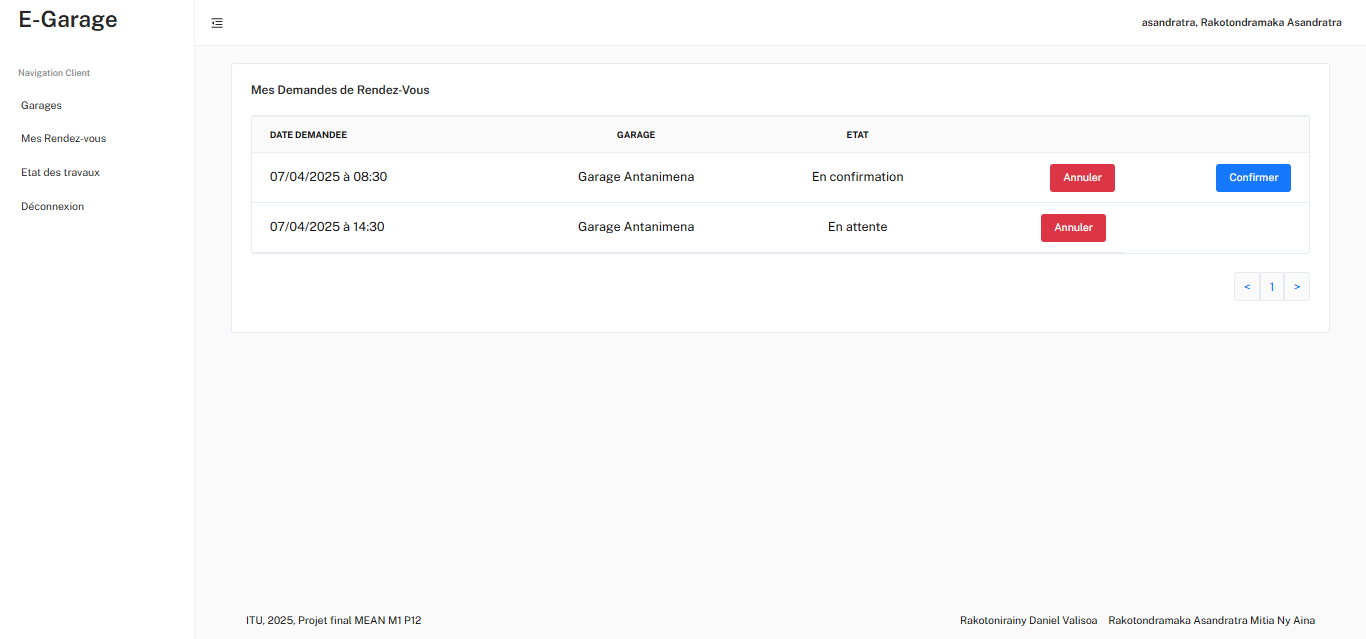
## Prise de rendez-vous

Cette fonctionnalité rassemble deux modules : la demande, et le suivi de rendez-vous.

### Demande de rendez-vous

Un client se connecte et vois la liste de garage. Il sélectionne le garage où il veut faire réparer sa voiture.

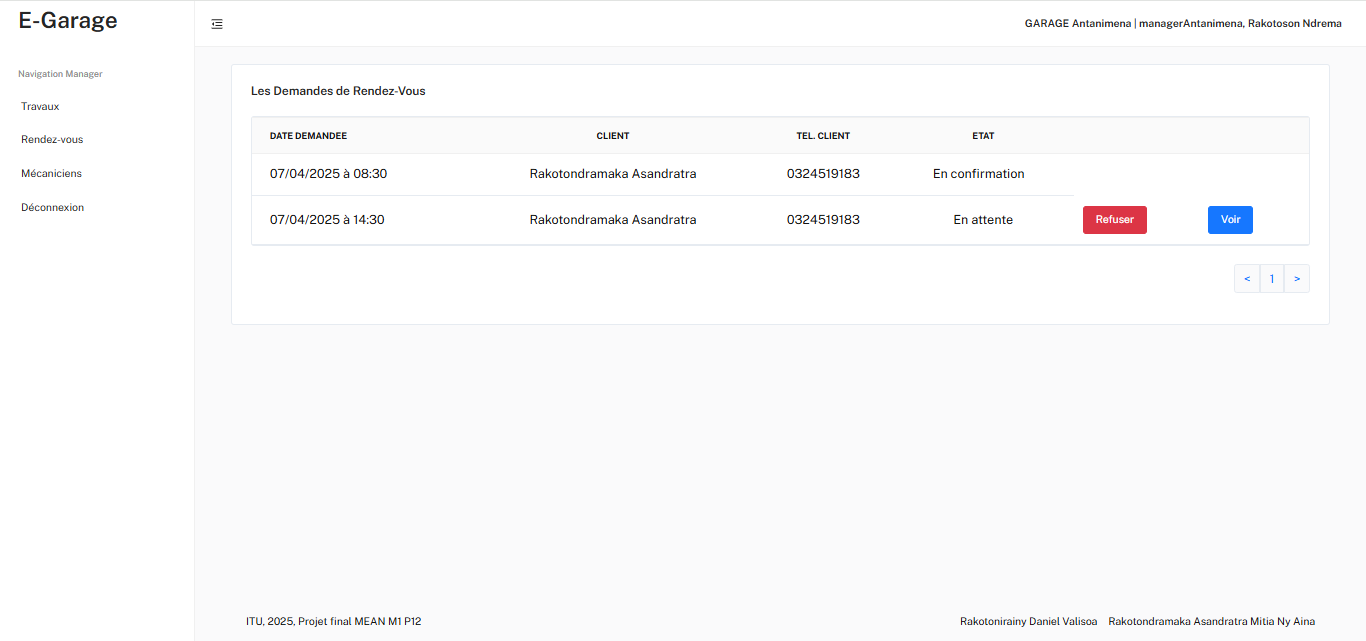
En sélectionnant un garage, il peut voir qui est le manager du garage et son contact, et il a également un formulaire pour envoyer une demande de rendez-vous, sur lequel le client mettra la date et l'heure demandée.

Après avoir envoyé sa demande, le client peut le voir en passant par la liste des rendez-vous, d'où il clique sur le lien 'mes demandes de rendez-vous', et il sera dirigé vers la liste de ses demandes de rendez-vous

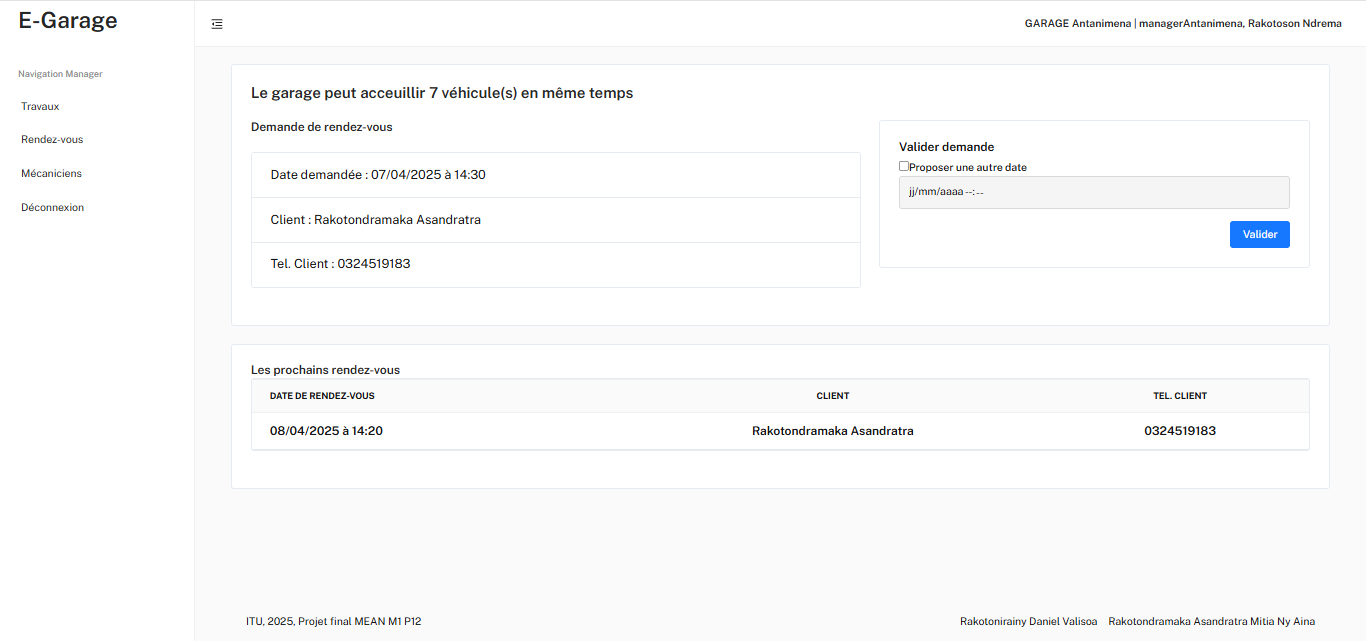
Cette liste de demandes présente l'état des demandes, soit elles sont en attente de réponse du garage, soit en attente de confirmation du client.

Si la demande est en confirmation, c'est qu'un manager a vu sa demande et souhaite la confirmation du client pour la date. En confirmant, le client crée le rendez-vous et a désormais un rendez-vous avec le garage.

Si la demande est en attente, c'est qu'il attend encore l'action du manager, dans ce cas-là, ça se passe comme suit :

Un manager se connecte, et va sur la page de liste de rendez-vous d'où il peut aller vers la page listant les demandes de rendez-vous envoyées au garage.

Le manager choisi une demande de rendez-vous et peut soit directement le refuser, soit aller vers le formulaire de validation de la demande.

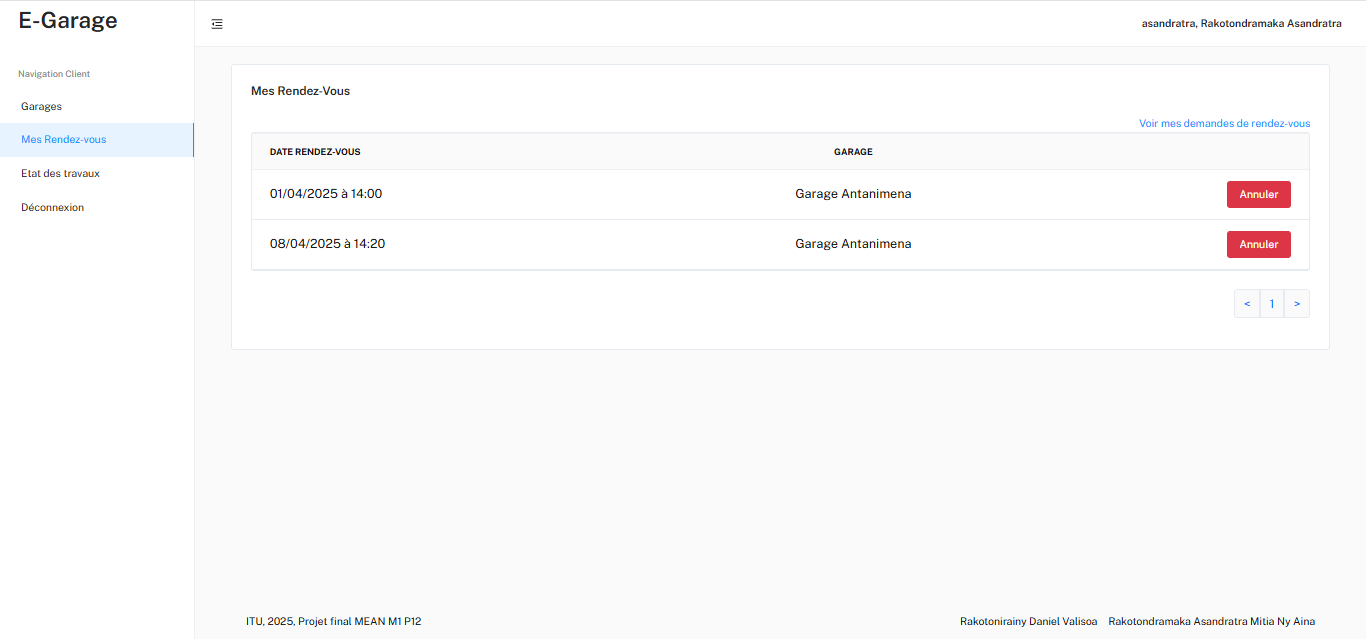
Depuis le formulaire de validation, il peut voir le demandant, la date, et les prochains rendez-vous du garage, pour voir la disponibilité.

Le manager peut modifier la date de rendez-vous.

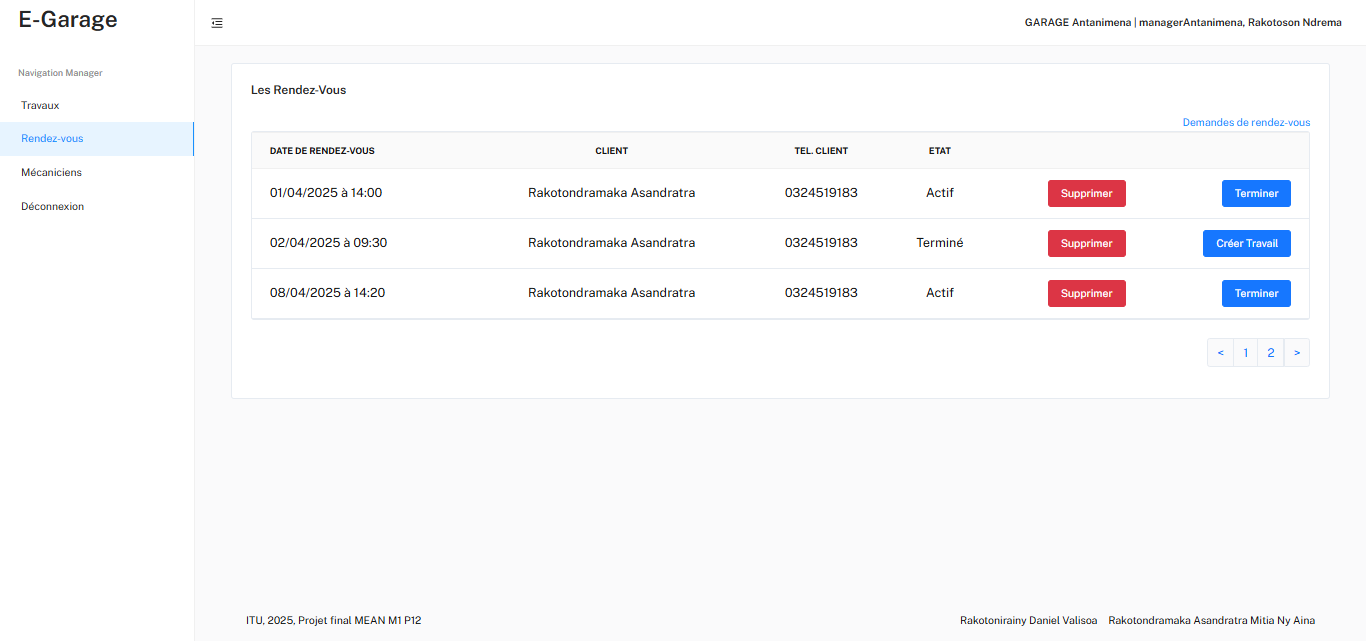
Après validation du manager, la demande sera en confirmation.

### Rendez-vous

Lorsque la demande de rendez-vous est validée par un manager et confirmer par le client, le rendez-vous sera enregistré.

Le client pourra voir la liste de ses rendez-vous encore actif et les annuler si besoin.

Le manager lui peut voir les rendez-vous de son garage, les terminer, et s’ils sont terminés, c'est que le rendez-vous s'est effectué.

Lorsqu'un rendez-vous est terminé, le manager peut directement créer un travail à partir de ce rendez-vous, travail qui sera rattaché au client.

Les rendez-vous restent visibles jusqu'à ce qu'ils soient supprimés par le manager.

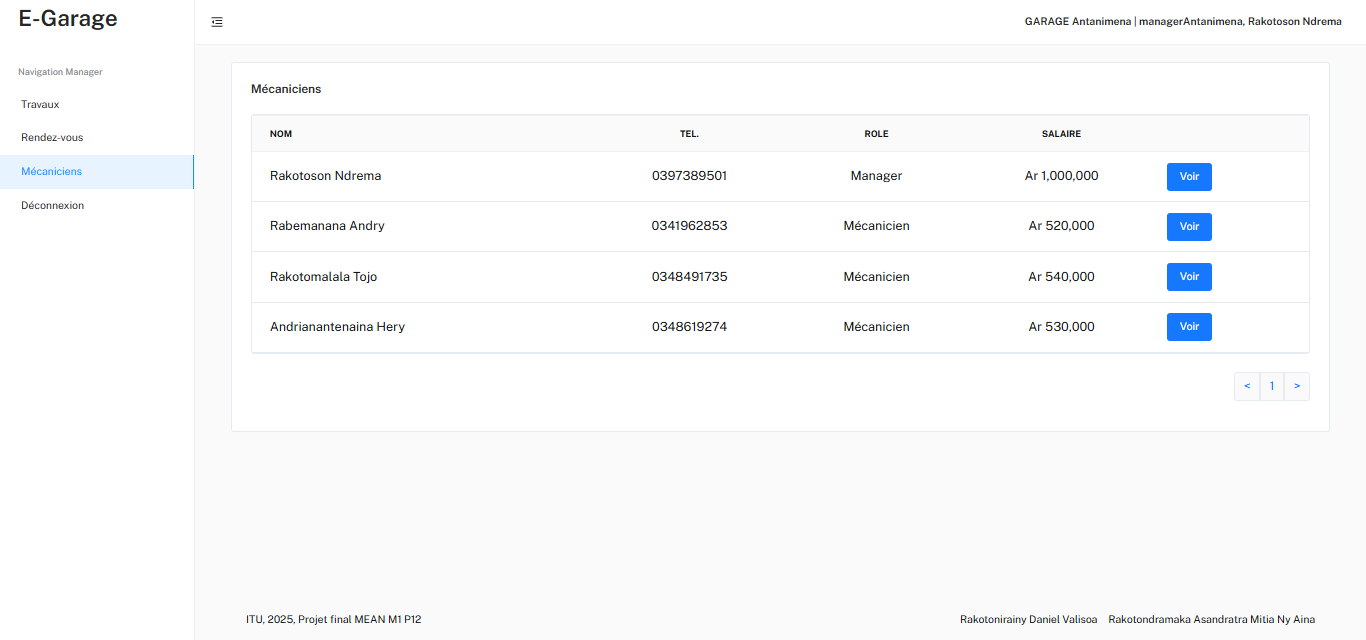
## Suivi de travail

## Suivi de paiement

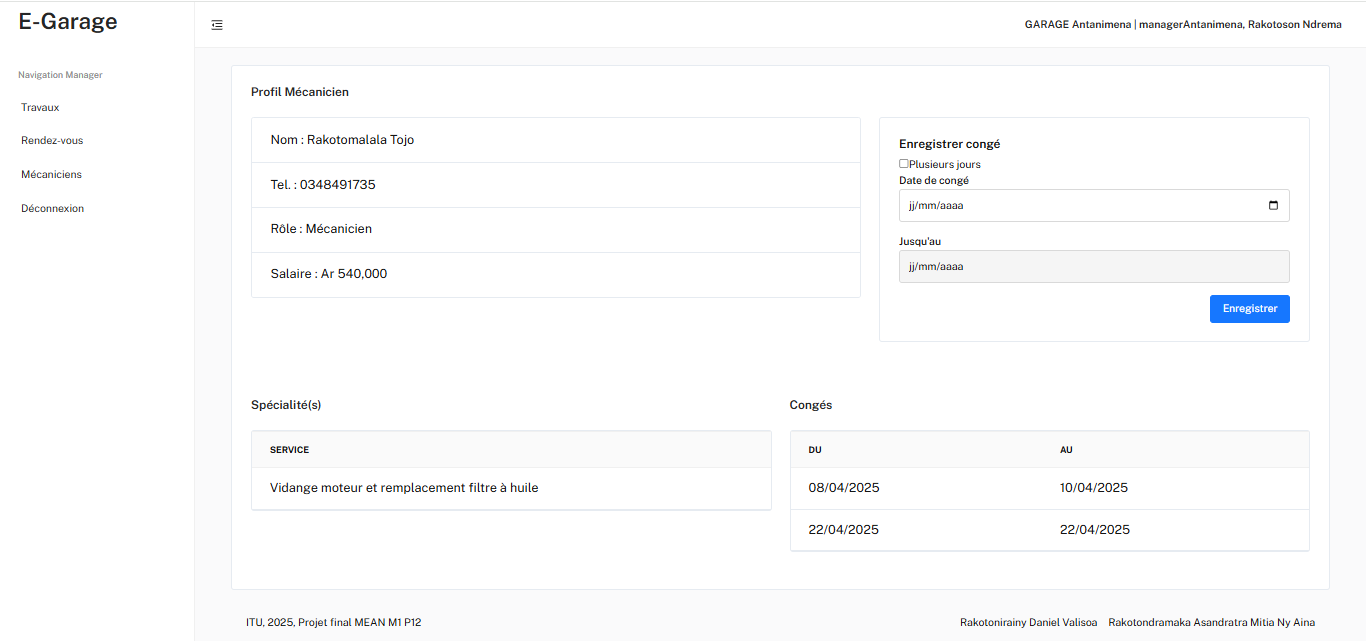
## Enregistrement de congé

Nous avons mis en place cette fonctionnalité pour donner une utilité en plus à l'utilisation de cette application pour le manager.

Le manager peut voir la liste des mécaniciens de son garages.



Depuis cette liste, il peut aller vers la page pour voir les détails des employés, leurs spécialités, leurs congés.



De cette page de détails d'employé, le manager peut enregistrer les congés de l'employé, pour un jour, ou un intervalle de jour.

Cette fonctionnalité ouvre une approche de gestion de ressources humaines à l'application.