

Garage Utility

Projet Java JSE - Swing

Marc Lauze

Sommaire

I.....	Contexte professionnelle
II.....	Description du projet
III.....	Spécificités de l'application
IV.....	Procédure d'utilisation
V.....	Technologies
VI.....	Base de données
VII.....	Amélioration a prévoir

I - Contexte professionnel

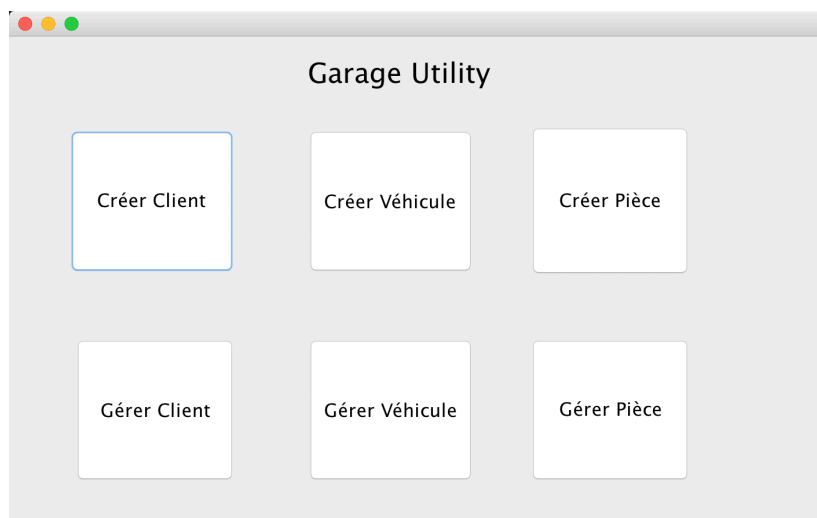
Les grandes franchises de garage en France disposent de logiciels de gestion de clients. Toutefois, les petits garages indépendants sont quand à eux forcés de payer des licences pour des logiciels de gestion de clients ou alors de ne pas informatiser leur fichiers clients. Garage Utility est une solution logicielle gratuite et open sources permettant à ces petits garages de gérer leur clientèle à moindre coût.

II - Présentation de l'application

Garage Utility est une application client lourd développée en Java JSE avec l'api Swing. Elle permet notamment aux petits garagistes indépendants d'avoir une solution logicielle pour leur gestion de clientèle. L'utilisateur peut enregistrer des informations sur ses clients, sur les véhicules sur lesquels il a travaillé mais aussi d'avoir un suivi des pièces commandées pour les différentes interventions.

III - Spécificités de l'application

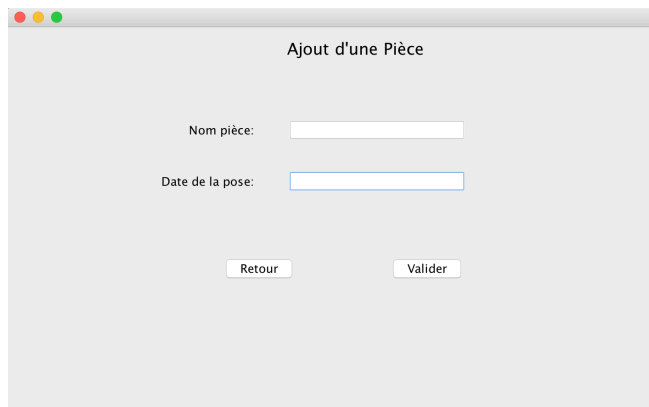
L'application se veut facile d'utilisation, elle donne accès à des fonctions de bases de gestion de clientèle, comme pouvoir consulter ou créer sa liste de clients, de pièces ou de véhicules. Epurée de tout superflu, l'application dispose donc d'un affichage clair des informations et des outils de navigation. Via une base de données, les données sont donc enregistrées et peuvent donc être consultées à la convenance de l'utilisateur.



Ci dessus, la fenêtre principale qui s'affiche lors du lancement de l'application

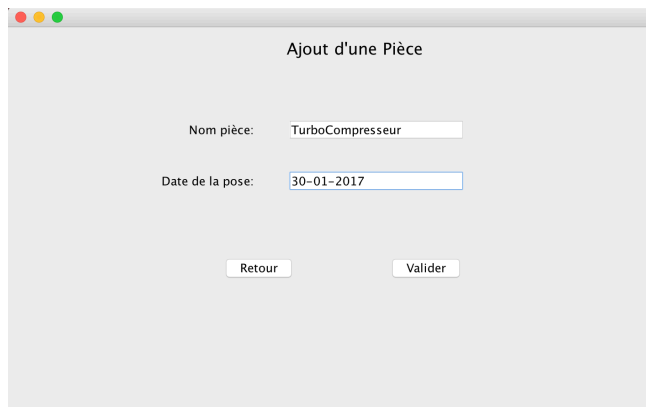
IV- Procédure d'utilisation

Ici, la procédure de création d'une pièce
(Identique a celle de la création d'un véhicule ou d'un client).



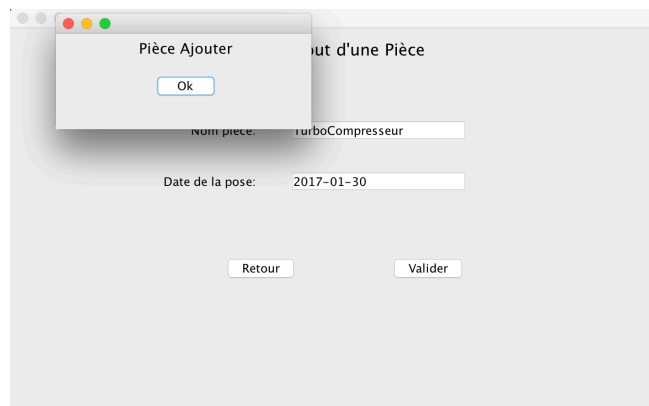
A screenshot of a macOS-style window titled "Ajout d'une Pièce". It contains two text input fields: "Nom pièce:" and "Date de la pose:". Below the fields are two buttons: "Retour" and "Valider".

figure 1



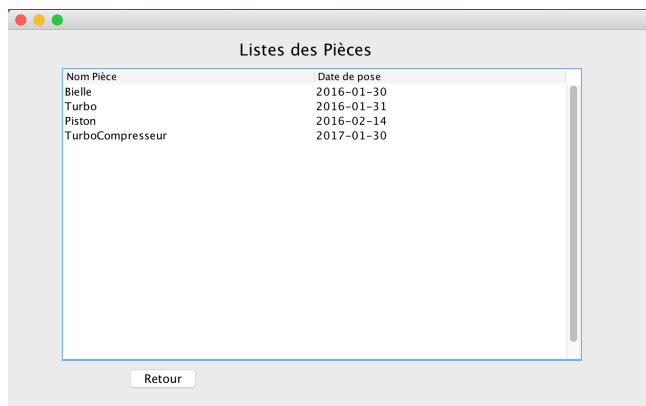
A screenshot of the "Ajout d'une Pièce" window. The "Nom pièce:" field is filled with "TurboCompresseur" and the "Date de la pose:" field is filled with "30-01-2017". The "Retour" and "Valider" buttons are still present.

figure 2



A screenshot of the "Ajout d'une Pièce" window. A small dialog box titled "Pièce Ajouter" with an "Ok" button is overlaid on top. The background window shows the "TurboCompresseur" piece and the date "2017-01-30".

figure 3



A screenshot of a window titled "Listes des Pièces". It contains a table with two columns: "Nom Pièce" and "Date de pose". The table lists four items: Bielle, Turbo, Piston, and TurboCompresseur, each with a corresponding date. A "Retour" button is at the bottom.

Nom Pièce	Date de pose
Bielle	2016-01-30
Turbo	2016-01-31
Piston	2016-02-14
TurboCompresseur	2017-01-30

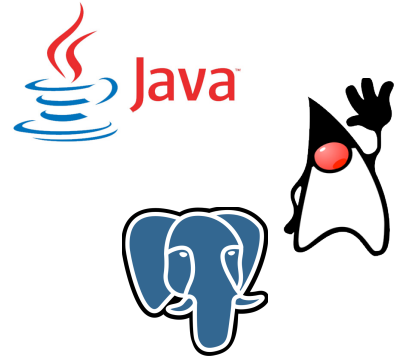
figure 4

La figure 1 représente la fenêtre de création d'une pièce, suivie de la procédure a suivre (figure 2 et 3) pour ajouter une pièce en base. Enfin la figure 4 présente le listing des pièces.

V - Technologies

L'application est développée à l'aide de Java JSE et de l'api Swing. Le choix de java est due à sa portabilité (Windows, Mac OS, Linux).

Le SGBD utilisé pour l'application est PostgreSQL, notamment grâce à son comportement stable mais aussi pour sa compatibilité avec le pilote JDBC de Java.



Au niveau des outils de développement, NetBeans a été l'IDE retenu pour le développement de l'application, due au comportement natif de Swing au sein de celui-ci. Il offre notamment l'avantage d'avoir une prévisualisation des fenêtres sans devoir lancer l'application.



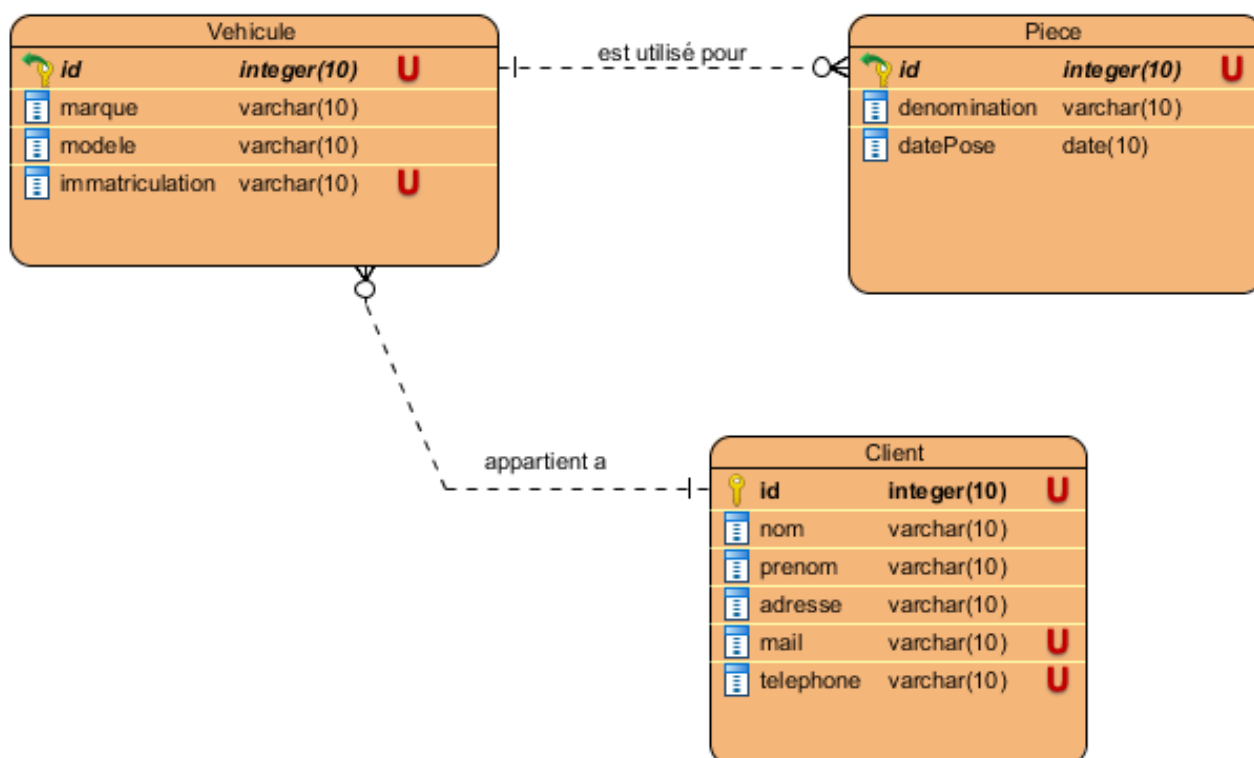
Afin d'avoir un suivi sur l'avancement du projet, le logiciel de versioning Git couplé avec le client GitHub a permis d'améliorer la productivité du développement de l'application.



VI - Base de données

La base de données est composé de 3 entités:

- Véhicule
- Pièce
- Utilisateur



La table Vehicule contient les informations sur le véhicules du client, comme la marque, le modèle et l'immatriculation. Ces informations permettent ainsi au garagiste de commander les pièces qui correspondent au véhicule à réparer. La table Piece, contient le nom de la pièce concerné ainsi que la date ou elle a été poser sur le véhicule afin d'assurer un suivis de l'usage de celle-ci. La table Client, quand à elle contient les informations nécessaire sur le client, soit, son nom, prénom, adresse, adresse mail et numéros de téléphone. Ces informations permettent au garagiste d'avoir un suivis sur sa clientèle.

VII - Améliorations à prévoir

Il est prévu dans l'avenir d'améliorer l'application afin de garantir une meilleure expérience utilisateur, comme la possibilité de modifier une entité directement depuis l'application, d'ajouter des filtres de recherches, comme par exemple par plaque d'immatriculation pour la liste des véhicules ou encore par nom de famille pour la recherche de client dans la base. Mais aussi créer un module pour la gestion des stocks des pièces du garagiste, afin d'être le plus réactif possible sur la vérification de la disponibilité de celles-ci.

Enfin, afin de mieux satisfaire la demande, il faudrait revoir les fonctionnalités listées ci-dessus et de penser "utilisateur" lors du développement de ces nouvelles fonctionnalités. En effet, une application est plébiscitée de par sa facilité d'utilisation, ce qui rejoint les objectifs de GarageUtility.