

# Gestion des pièces d'un garage

## Table des matières

1.	Introduction .....	1
2.	Description du besoin.....	1
3.	Travail à réaliser.....	3

## 1. Introduction

Il est proposé de travailler sur la modélisation d'un logiciel de gestion des pièces d'un garage automobile, garage concessionnaire d'une marque.

L'objectif n'est pas de faire une analyse complète du logiciel, mais de réaliser les principaux schémas UML.

Il n'y a pas de solution technique unique pour répondre à un besoin. Cette correction est donc donnée à titre informatif, d'autres solutions sont envisageables.

## 2. Description du besoin

Le logiciel doit permettre de gérer les pièces pour la réparation des voitures, ce logiciel est utilisé par le mécanicien, le chef d'atelier, le comptable et le responsable du garage.

À terme, les clients doivent pouvoir se connecter au logiciel via internet afin de connaître les délais de commande des pièces. Pour éviter que la concurrence ne voie les prix proposés par le garage, les clients n'auront pas accès par ce biais aux prix des pièces.

Le mécanicien doit pouvoir utiliser le logiciel dans l'atelier au moyen d'un écran tactile pouvant être nettoyé.

Le système doit utiliser un service d'authentification pour savoir si l'utilisateur est déclaré dans le système et quels sont ses droits. Pour pouvoir utiliser l'application, tous les utilisateurs sont obligés de se connecter au moyen d'un identifiant et d'un mot de passe.

Le système se connecte régulièrement à la base des pièces de la marque afin de télécharger les nomenclatures des pièces des différents modèles. Cette opération de mise à jour peut-être déclenchée manuellement par le chef d'atelier. Les nomenclatures sont alors stockées localement. Le système de stockage doit pouvoir contenir entre 1 000 000 et 1 500 000 pièces avec le schéma de montage correspondant.

Le système doit générer des logs, logs qui sont analysées par un système d'audit et de rapports.

Le système doit permettre de générer les factures de commandes de pièces, le contenu de la facture doit pouvoir être configurable (logo, numero siret etc.) et la facture doit pouvoir avoir différents formats : word ; html, pdf etc.

Une pièce peut être référencée sans jamais avoir été commandée, en stock, épuisée, en commande, arrivée au service de réception, rangée. Le mécanicien lorsqu'il a besoin d'une pièce peut voir si elle est référencée, en stock, épuisée, ou en commande. Lorsqu'une pièce est en commande, le chef d'atelier peut voir si la demande de pièce a été reçue, si elle est en fabrication ou envoyée. Si une pièce a été demandée depuis plus d'un mois elle est considérée comme épuisée.

Une pièce qui n'est pas en stock peut être commandée par le mécanicien, le mécanicien doit en faire la demande au chef d'atelier. Pour effectuer une commande, le chef d'atelier doit regarder régulièrement la liste des demandes de commande, et sélectionner une demande. Puis il effectue la recherche d'un fournisseur de la pièce, si la pièce est référencée par le constructeur, le chef d'atelier peut la commander. Si le chef d'atelier trouve un autre fournisseur, le chef d'atelier effectue une demande auprès du service comptable qui doit autoriser la commande après avoir vérifié que la pièce n'est pas référencée chez le constructeur. La nouvelle référence fait partie intégrante du système. Si aucun fournisseur n'est trouvé, le chef d'atelier informe le mécanicien.

Toutes les pièces ont un nom, une référence, un schéma de montage, un prix. Les pièces mécaniques ont un poids et un encombrement ; les pièces de tôlerie ont une surface, une résistance mécanique ; les pièces plastiques une élasticité.

Les pièces de fournisseurs autres que le constructeur ont de plus les références du fournisseur.

Le logiciel doit pouvoir fonctionner sur les Os les plus courants du marché.

### **3. Travail à réaliser**

#### **1. Cas d'utilisation**

1.1. Faire un schéma avec les acteurs

1.2. Réaliser un schéma de contexte en utilisant un schéma de cas d'utilisation

1.3. Réaliser un schéma de cas d'utilisation pour l'acteur chef d'atelier

#### **2. Diagramme d'activité**

2.1. Réaliser le diagramme d'activité de la commande de pièce par le chef d'atelier. Ce premier schéma ne contient qu'un couloir et ne modélise pas les données utilisées dans l'activité

2.2. Réaliser le diagramme d'activité de la connexion d'un utilisateur à l'application. Ce diagramme doit contenir deux couloirs un pour l'acteur et un pour le système, ce diagramme doit visualiser les données échangées ; et ce diagramme doit modéliser au moins une exception. Pendant que l'application vérifie les données de connexion de l'utilisateur, un écran de « publicité » est affiché à l'utilisateur.

#### **3. Diagrammes de classes**

3.1. Réaliser le diagramme de classes des pièces, a pour objectif de modéliser les classes avec éventuellement leurs attributs pas les méthodes

3.2. Enrichir le diagramme de classes des pièces afin de pouvoir gérer les pièces par type, sous-types, sous-sous-types etc.

3.3. Réaliser un premier diagramme de classes des factures

3.4. Enrichir ce diagramme des factures afin de modéliser une interface d'accès aux factures

3.5. Enrichir ce deuxième diagramme des factures pour séparer les aspects de format des factures des aspects données des factures

## 4. Diagrammes d'états

4.1. Faire la liste des états d'une pièce

4.2. Définir les macros états et les transitions s'appliquant à plusieurs états

4.3. Créer un diagramme d'états en positionnant les états sans mettre les transitions

4.4. Enrichir le diagramme en créant les transitions

4.5. Enrichir le diagramme par les gardes de certaines transitions

4.6. Modifier le diagramme pour que le chef d'atelier soit automatiquement prévenu quand une pièce est rangée

## 5. Diagramme de séquences

5.1. Réaliser le diagramme séquence de commande d'une pièce par le chef d'atelier

5.1.1. Lorsque cela est nécessaire, ajouter des méthodes aux classes existantes

5.1.2. Lorsque cela est nécessaire, ajouter les classes manquantes

## 6. Diagramme de packages

6.1. Réaliser le diagramme de package de l'application

6.2. Répartir les classes dans les packages

## 7. Diagramme de déploiement

7.1. Réaliser le diagramme de déploiement de l'application