

GSBCAR – Compte-rendu

Pour ce premier projet personnel encadré, nous nous sommes glissés dans la peau d'employés d'une entreprise de développement de logiciels professionnels. Nous avons reçu une demande d'une entreprise externe (GSB) qui souhaitait posséder une application de gestion de leurs véhicules de fonction avec d'un côté une partie pour un technicien, et une autre partie pour les employés. La partie effectuée lors de ce PPE concerne le technicien.

Afin de répondre aux exigences de cette entreprise, nous avons du passer par plusieurs étapes :

- *Création du diagramme des cas d'utilisation.*
- *Création du diagramme UML.*
- *Mise en forme de l'application.*
- *Création du MCD.*
- *Création de la base de données.*
- *Création du code et lien avec la base de données.*

Nous allons détailler celles-ci au cours de ce rapport afin d'expliquer comment nous sommes parvenus à ce résultat :

1. Diagramme des cas d'utilisation :



2. Design Windows Form :

A – Connexion

Design de la fenêtre de connexion (Form1) pour GSBCAR 1.0. La fenêtre contient :

- Le titre "GSBCAR 1.0" en gros caractères.
- Des champs de saisie pour "Identifiant :" et "Mot de passe :".
- Un bouton "Valider" pour soumettre les informations.
- Un bouton "Quitter" en bas à droite pour fermer l'application.

Ici nous avons créé la première page qui est la page de connexion, avec possibilité de quitter l'application. Chaque utilisateur sera bien entendu inclus dans la base de données.

B – Menu

The 'Menu' window has a title bar with standard Windows controls. The main area is titled 'Liste des voitures :'. It contains a large empty list box labeled 'listBox1' and a label 'Etat :'. At the bottom, there are two buttons: 'Ajouter voiture' and 'Déconnexion'.

Une fois que le technicien est connecté, il a accès au menu principal ou il pourra avoir accès à toutes les voitures ainsi qu'à leur marque, modèle, id et leur état, il pourra en ajouter une ou bien se déconnecter. Pour voir son état, n'a qu'à cliquer une fois sur le véhicule, pour les détails il devra faire un double clic.

C – Détails

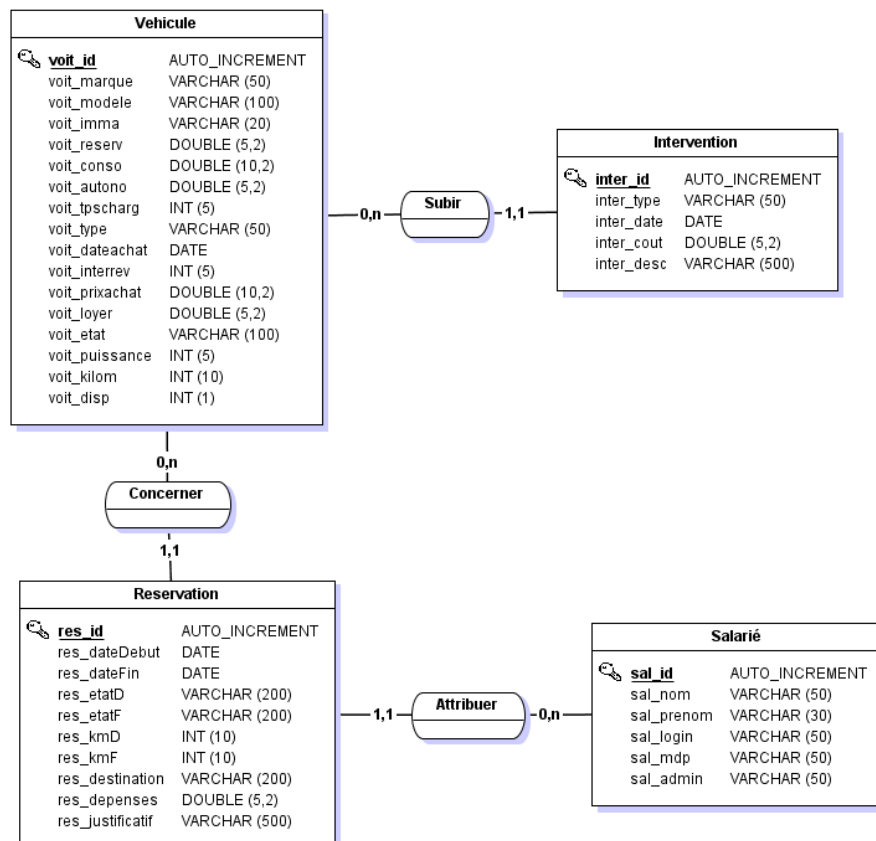
The 'Details' window has a title bar with standard Windows controls. The main area is divided into several sections. On the left, there is a list of vehicle attributes, each followed by an input field: 'Type', 'Marque', 'Modèle', 'Immatriculation', 'Puissance', 'Kilométrage', 'Consommation', 'Autonomie', 'Temps de chargement', 'Date d'achat', 'Prix d'achat' (with a '€' symbol), 'Loyer' (with a '€' symbol), 'Etat', and 'Nombre d'interventions'. In the center, there is a section titled 'Historique des interventions :'. It contains a table with four columns: 'Date', 'Type', 'Description', and 'Prix'. Each column has a corresponding list box labeled 'listBox1', 'listBox2', 'listBox3', and 'listBox4' respectively. On the right side, there are four buttons: 'Déclarer un retour', 'Déclarer une indisponibilité', 'Modifier voiture', and 'Supprimer voiture'. At the bottom right, there is a 'Retour' button.

Une fois que le technicien est redirigé sur la page des détails, il aura accès à toutes les informations importantes de la voiture concernée ainsi qu'à son historique des différentes interventions. Il pourra s'il le souhaite les modifier ou même supprimer la voiture (celle-ci ne sera qu'archivée si elle a subi au moins une intervention), il pourra de plus déclarer une indisponibilité si besoin et déclarer son retour lorsqu'elle aura terminé son indisponibilité.

D – Indisponibilité

S'il veut déclarer une indisponibilité, le technicien aura accès à une page qui lui demandera de préciser le type d'indisponibilité, le coût de celle-ci et la date.

3. Le MCD :



4. La base de données SQL :


intervention

reservation

salarie

vehicule

A – Intervention

inter_id		int(11)
inter_type		varchar(50) latin1_swedish_ci
inter_date		date
inter_cout		double
voit_id		int(11)
inter_desc		varchar(500) latin1_swedish_ci


B – Reservation

res_id		int(11)
res_dateDebut		date
res_dateFin		date
res_etatD		varchar(50) latin1_swedish_ci
res_etatF		varchar(50) latin1_swedish_ci
res_kmD		int(11)
res_kmF		int(11)
res_destination		varchar(100) latin1_swedish_ci
res_depenses		double
res_justificatif		varchar(100) latin1_swedish_ci
sal_id		int(11)
voit_id		int(11)

C – Salarié

sal_id		int(11)
sal_nom		varchar(50) latin1_swedish_ci
sal_prenom		varchar(50) latin1_swedish_ci
sal_login		varchar(50) latin1_swedish_ci
sal_mdp		varchar(50) latin1_swedish_ci
sal_admin		varchar(3) latin1_swedish_ci

D – Vehicule

voit_id 	int(11)
voit_marque	varchar(50) latin1_swedish_ci
voit_modele	varchar(50) latin1_swedish_ci
voit_imma	varchar(15) latin1_swedish_ci
voit_conso	double
voit_autono	double
voit_tpscharg	int(11)
voit_type	varchar(20) latin1_swedish_ci
voit_dateachat	date
voit_interrev	int(11)
voit_prixachat	double
voit_loyer	double
voit_etat	varchar(30) latin1_swedish_ci
voit_puissance	int(5)
voit_kilom	int(10)
voit_disp	int(1)

Conclusion :

Pour la réalisation de cette application nous avons mis environ 25 heures au total en comprenant toutes les étapes. Il y eu d'abord une grande session d'analyse qui reste importante à maîtriser afin de mieux travailler par la suite. Celle-ci fut suivie de la mise en forme des différentes données qui allaient être utilisées puis nous avons terminé sur la construction du code afin de faire interagir l'application avec la base de données. Ce PPE ne nous a pas foncièrement posé de problème, hormis qu'il nous a juste fallu un temps d'adaptation pour comprendre comment fonctionnait le système de requêtes MySql en C# qui était la grande nouveauté à appliquer, mais une fois cela fait le code avançait bien.

Cette expérience nous a permis d'avoir un léger aperçu de ce que représente le travail en entreprise et de commencer à nous y préparer, ainsi que de nous apprendre à réfléchir sur un projet donné en un temps limité, donc à optimiser celui-ci et augmenter notre réactivité.