

# Projet

## Rent A Car, Agence de location de voitures



## SOMMAIRE

I – Modèle conceptuel et logique de donnée (MCD & MLD) -- 3

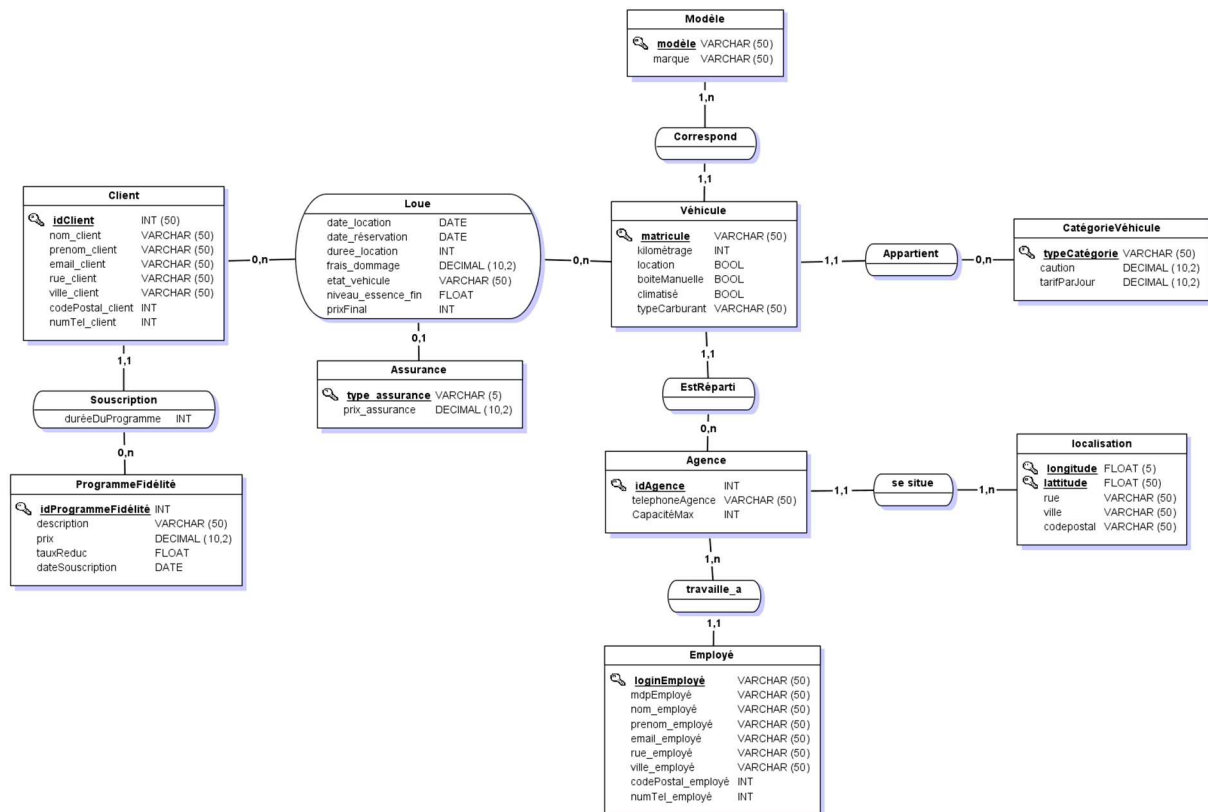
II – Script SQL ----- 5

III – Graphical User Interface ----- 7

IV – Requêtes ----- 9

## I – Modèle conceptuel et logique de donnée (MCD & MLD)

Selon l'énoncé du projet que nous avons reçu, nous avons créé le **MCD** suivant en prenant en compte les particularités de chaque entités, ainsi que les liens qui pouvaient les unir. Voici le **MCD** :



Il y a 7 entités au total. Nous avons décidé de ne pas utiliser l'héritage car cela aurait posé des problèmes au niveau du script sql de la base de donnée. Nous avons également décidé de rajouter une entité **Localisation** à part afin que dans notre **MLD** nous n'ayons pas besoin de séparer l'entité **Agence** pour respecter la 3e forme normale (3FN).

Enfin, nous avons établi une association ternaire **Loue** qui a comme attribut plusieurs caractéristiques (niveau\_essence\_fin, frais\_dommage...). Dans le **MLD**, celle-ci deviendra relation et donc une table dans notre base de donnée.

Grâce à **JMerise**, nous générons le **MLD** suivant :



Comme dit précédemment, l'association **loue** est devenue relation est a comme clé primaire étrangère les relations à laquelle elle était liée (ici **Client**, **Véhicule** et **Assurance**). C'est cette relation qui nous permettra de faire le lien pour la location de véhicule par des clients de RentACar.

Enfin, toutes les relations sont en **3FN**, c'est-à-dire que chaque attribut est atomique (non composé et mono-valué), chaque attribut non-clé dépend uniquement de la totalité de la clé et non d'une partie et ne dépend pas d'autres attribut non-clé.

## II – Script SQL

Le script utilisé est celui généré par JMerise :

```
#-----  
#      Script MySQL.  
#-----  
  
#-----  
# Table: ProgrammeFidelite  
#-----  
  
CREATE TABLE ProgrammeFidelite(  
  idProgrammeFidelite Int NOT NULL ,  
  description Varchar (50) NOT NULL ,  
  prix Decimal (10,2) NOT NULL ,  
  tauxReduc Float NOT NULL ,  
  dateSouscription Date NOT NULL  
  ,CONSTRAINT ProgrammeFidelite_PK PRIMARY KEY (idProgrammeFidelite)  
)ENGINE=InnoDB;  
  
#-----  
# Table: Client  
#-----  
  
CREATE TABLE Client(  
  idClient Int NOT NULL ,  
  nom_client Varchar (50) NOT NULL ,  
  prenom_client Varchar (50) NOT NULL ,  
  email_client Varchar (50) NOT NULL ,  
  rue_client Varchar (50) NOT NULL ,  
  ville_client Varchar (50) NOT NULL ,  
  codePostal_client Int NOT NULL ,  
  numTel_client Int NOT NULL ,  
  dureeDuProgramme Int NOT NULL ,  
  idProgrammeFidelite Int NOT NULL  
  ,CONSTRAINT Client_PK PRIMARY KEY (idClient)  
  ,CONSTRAINT Client_ProgrammeFidelite_FK FOREIGN KEY (idProgrammeFidelite) REFERENCES ProgrammeFidelite(idProgrammeFidelite)  
)ENGINE=InnoDB;  
  
#-----  
# Table: localisation  
#-----  
  
CREATE TABLE localisation(  
  longitude Float NOT NULL ,  
  latitude Float NOT NULL ,  
  rue Varchar (50) NOT NULL ,  
  ville Varchar (50) NOT NULL ,  
  codepostal Varchar (50) NOT NULL  
  ,CONSTRAINT localisation_PK PRIMARY KEY (longitude,latitude)  
)ENGINE=InnoDB;  
  
#-----  
# Table: Agence  
#-----  
  
CREATE TABLE Agence(  
  idAgence Int NOT NULL ,  
  telephoneAgence Varchar (50) NOT NULL ,  
  capaciteMax Int NOT NULL ,  
  longitude Float NOT NULL ,  
  latitude Float NOT NULL  
  ,CONSTRAINT Agence_PK PRIMARY KEY (idAgence)  
  ,CONSTRAINT Agence_localisation_FK FOREIGN KEY (longitude,latitude) REFERENCES localisation(longitude,latitude)  
)ENGINE=InnoDB;
```

```
#-----
# Table: Véhicule
#-----

CREATE TABLE Vehicule(
  matricule      Varchar (50) NOT NULL ,
  kilometrage    Int NOT NULL ,
  location       Bool NOT NULL ,
  boiteManuelle  Bool NOT NULL ,
  climatise      Bool NOT NULL ,
  typeCarburant  Varchar (50) NOT NULL ,
  typeCategorie  Varchar (50) NOT NULL ,
  idAgence       Int NOT NULL ,
  modele         Varchar (50) NOT NULL
  ,CONSTRAINT Vehicule_PK PRIMARY KEY (matricule)

  ,CONSTRAINT Vehicule_CategorieVehicule_FK FOREIGN KEY (typeCategorie) REFERENCES CategorieVehicule(typeCategorie)
  ,CONSTRAINT Vehicule_Agence0_FK FOREIGN KEY (idAgence) REFERENCES Agence(idAgence)
  ,CONSTRAINT Vehicule_Modele1_FK FOREIGN KEY (modele) REFERENCES Modele(modele)
)ENGINE=InnoDB;

#-----
# Table: Employé
#-----

CREATE TABLE Employe(
  loginEmploye   Varchar (50) NOT NULL ,
  mdpEmploye     Varchar (50) NOT NULL ,
  nom_employe    Varchar (50) NOT NULL ,
  prenom_employe Varchar (50) NOT NULL ,
  email_employe  Varchar (50) NOT NULL ,
  rue_employe    Varchar (50) NOT NULL ,
  ville_employe  Varchar (50) NOT NULL ,
  codePostal_employe Int NOT NULL ,
  numTel_employe Int NOT NULL ,
  idAgence       Int NOT NULL
  ,CONSTRAINT Employe_PK PRIMARY KEY (loginEmploye)

  ,CONSTRAINT Employe_Agence_FK FOREIGN KEY (idAgence) REFERENCES Agence(idAgence)
)ENGINE=InnoDB;

#-----
# Table: Loue
#-----

CREATE TABLE Loue(
  idClient      Int NOT NULL ,
  matricule      Varchar (50) NOT NULL ,
  type_assurance Varchar (5) NOT NULL ,
  date_location  Date NOT NULL ,
  date_reservation Date NOT NULL ,
  duree_location Int NOT NULL ,
  frais_dommage  Decimal (10,2) NOT NULL ,
  etat_vehicule  Varchar (50) NOT NULL ,
  niveau_essence_fin Float NOT NULL
  ,CONSTRAINT Loue_PK PRIMARY KEY (idClient,matricule,type_assurance)

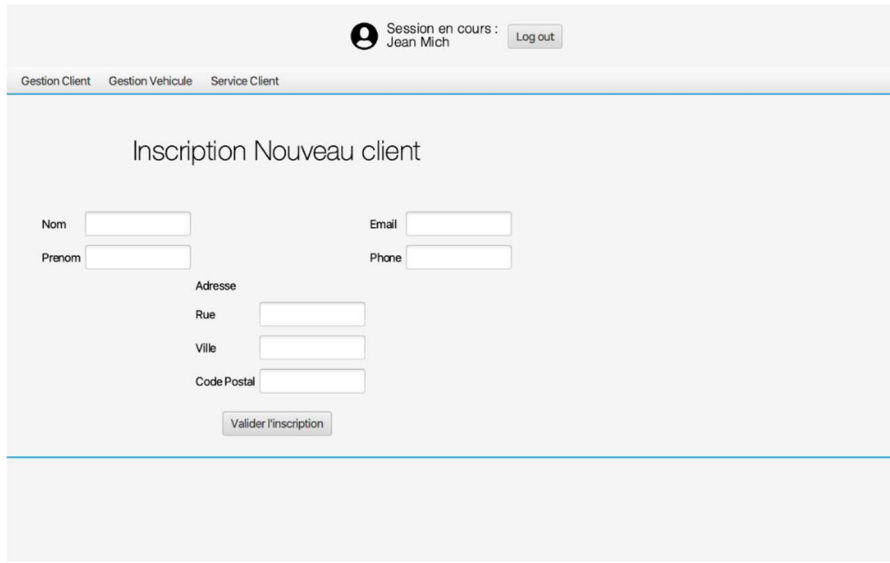
  ,CONSTRAINT Loue_Client_FK FOREIGN KEY (idClient) REFERENCES Client(idClient)
  ,CONSTRAINT Loue_Vehicule0_FK FOREIGN KEY (matricule) REFERENCES Vehicule(matricule)
  ,CONSTRAINT Loue_Assurance1_FK FOREIGN KEY (type_assurance) REFERENCES Assurance(type_assurance)
)ENGINE=InnoDB;
```

Nous avons rajouté quelques lignes au début du script pour que cela se mette directement dans MySQL:

```
DROP DATABASE IF EXISTS RentACar;
CREATE DATABASE RentACar;
USE RentACar;
```

### III – Graphical User Interface (interface application)

L'interface a été réalisé à l'aide de **Java** et du langage **XML**, voici un exemple de quelques visuels de notre interface :

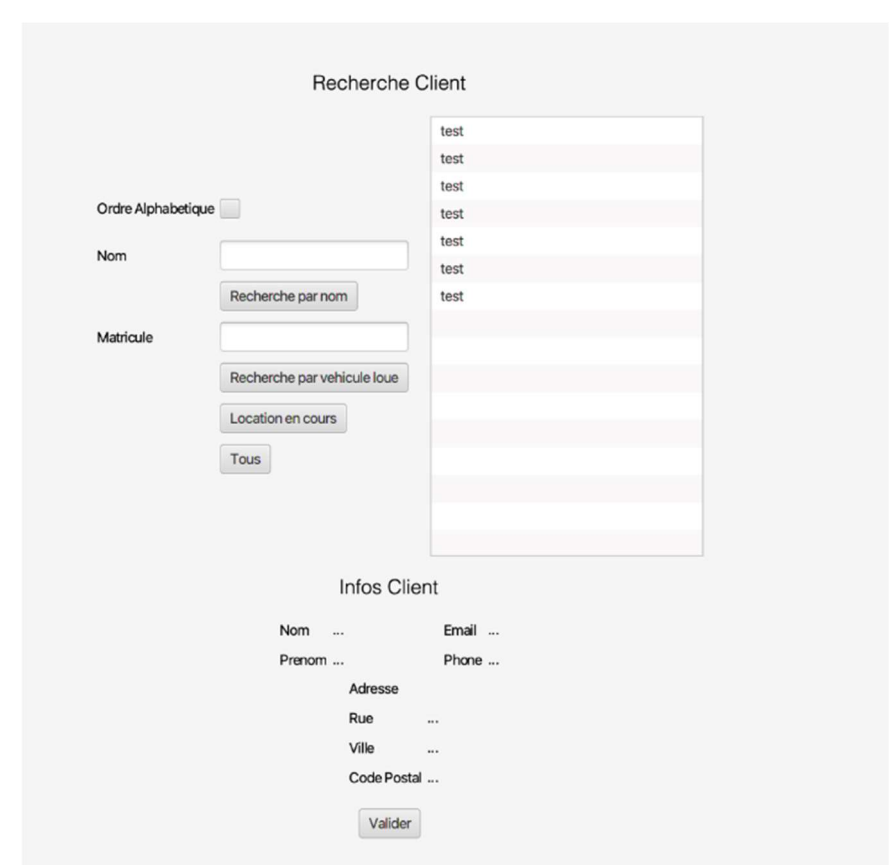


Session en cours : Jean Mich [Log out](#)

[Gestion Client](#) [Gestion Vehicule](#) [Service Client](#)

### Inscription Nouveau client

Nom  Email   
Prenom  Phone   
Adresse  
Rue   
Ville   
Code Postal   
[Valider l'inscription](#)



### Recherche Client

Ordre Alphabetique ☐

Nom  [Recherche par nom](#)

Matricule  [Recherche par vehicule loue](#)  
[Location en cours](#)  
[Tous](#)

test
test
test
test
test
test
test

### Infos Client

Nom ... Email ...  
Prenom ... Phone ...  
Adresse  
Rue ...  
Ville ...  
Code Postal ...  
[Valider](#)

### Modifier infos client

Nom	<input type="text"/>	Email	<input type="text"/>
Prenom	<input type="text"/>	Phone	<input type="text"/>
Adresse			
Rue	<input type="text"/>		
Ville	<input type="text"/>		
Code Postal	<input type="text"/>		
<input type="button" value="Valider modification"/>			

## Ajout Nouveau Vehicule

Matricule	<input type="text"/>
Kilometrage	<input type="text"/>
Matricule	<input type="text"/>
Modele	<input type="text"/>
Marque	<input type="text"/>
Climatisation	<input type="checkbox"/>
Boite Vitesse Manuelle	<input type="checkbox"/>
Categorie	<input type="button" value="Economique ▼"/>
<input type="button" value="Valider l'ajout"/>	



## IV – Requêtes

Pour modifier la base de donnée depuis notre application, nous avons utiliser l'**API JDBC**, qui nous permet d'envoyer des requêtes **SQL**. Nous avons donc créé différentes requêtes, voici les principales:

- **AddClient**, cette méthode permet d'ajouter un nouveau client avec un nouvel idClient, son nom, son prénom...
- **AddCar**, cette méthode permet d'ajouter un nouveau véhicule avec son matricule, son kilometrage...
- **DeleteClient** et **DeleteCar** permettent d'enlever un client, un véhicule de la base données selon, respectivement, un idClient et un matricule
- **EditClient** et **EditCar** permettent de changer les informations d'un client et d'une voiture dans la base de données
- **FindClient** et **FindCar** permettent de faire un **SELECT** d'un client, d'un véhicule selon des critères
- **ValidLocation** permet de créer une Devis dans la table Loue avec les informations suivantes : idClient, matricula, type\_assurance, date\_location