

## Mini-projet N°3 : Conception et réalisation d'une application de gestion des locations de véhicules

### **Enoncé:**

Une agence de location de véhicules souhaite informatiser la gestion de ses activités dans le but d'améliorer le rendement du personnel. Faciliter la procédure de réservation et location pour les clients. En outre, minimiser l'utilisation du papier et pouvoir effectuer des analyses statistiques en exploitant les données numériques de l'agence. L'objet du présent cahier des charges (version résumée) est la fourniture d'une application destinée à gérer toutes les activités de cette agence.

Dans ce but, on se propose dans ce mini-projet **d'analyser, faire la conception et réaliser une application desktop** permettant de gérer les activités de cette agence de location de véhicules. Votre Application Java devra proposer à l'utilisateur un menu permettant de naviguer dans les différentes fonctionnalités offertes par l'application (voir ci-dessous).

#### **(1) Gestion des utilisateurs de l'application**

Ce module sera géré par l'administrateur seulement pour lui permettre de créer de nouveaux utilisateurs de l'application. En plus, modifier les informations d'un utilisateur, suspendre le compte d'un utilisateur pendant sa période de congé et supprimer un utilisateur. En outre, l'administrateur de l'application est un utilisateur privilégié, qui peut faire appel à tous les services offerts par l'application.

#### **(2) Gestion des clients**

Ce module permettra de :

- Ajouter un nouveau client avec toutes ses informations qui vous semblent utiles (code\_client, nom complet, Adresse, N° GSM, image scannée de son permis de conduire, ...etc.);
- Modifier, ou supprimer un client existant spécifié par un nom ;
- Afficher la liste des clients par ordre alphabétique ;
- Rechercher un client par son nom complet et affiche l'enregistrement correspondant.
- Retour au menu principal.

#### **(3) Gestion des réservations**

Les réservations par téléphone ne peuvent être effectuées que par les clients de l'agence et non pas par un nouveau client. Un nouveau client doit se présenter à l'agence pour pouvoir effectuer une réservation ou signé directement un contrat de location après son ajout comme étant client dans l'application.

Ce module permettra de :

- Ajouter une nouvelle réservation avec toutes ses informations qui vous semblent utiles (code\_réservation, date réservation, date départ, date retour,...etc.) ;
- Modifier, ou supprimer une réservation existante ;
- Afficher la liste des réservations par ordre décroissant de la date de réservation ;

- Afficher la liste des réservations validées ;
- Afficher la liste des réservations non validées ;
- Afficher la liste des réservations annulées ;
- Rechercher une réservation par son code et affiche l'enregistrement correspondant ;
- Retour au menu principal.

#### **(4) Gestion des contrats de location:**

Une réservation validée avant 2 jours de la date de location du véhicule donnera lieu à un nouveau contrat. Le client doit restituer le véhicule qu'il a loué le jour de l'échéance du contrat de location. Chaque jour de retard est sanctionné par une amende de 2000 DH.

Ce module permettra de :

- Ajouter un nouveau contrat avec toutes ses informations qui vous semblent utiles (N° contrat; date\_contrat; date \_échéance; ...etc) ;
- Modifier, ou supprimer un contrat spécifié par code contrat ;
- Afficher la liste des contrats par ordre décroissant de la date du contrat ;
- Rechercher un contrat par son code et affiche l'enregistrement correspondant ;
- Retour au menu principal.

#### **(5) Gestion des factures**

Une facture concerne un seul contrat et sera livré au client après édition et signature du contrat par ce dernier.

Ce module permettra de :

- Ajouter une nouvelle facture avec toutes ses informations qui vous semblent utiles (N° facture ; date de facture; montant a payé, ...etc.) ;
- Modifier, ou supprimer une facture existante ;
- Afficher la liste des factures par ordre décroissant de la date de facture ;
- Rechercher une facture par son code et affiche l'enregistrement correspondant ;
- Retour au menu principal.

#### **(6) Gestion des véhicules**

Ce module permettra de :

- Ajouter un nouveau véhicule avec toutes ses informations qui vous semblent utiles (N°\_Immatriculation ; marque ; type; carburant ; compteur de KM, date de mise en circulation ; ...etc.) ;
- Modifier, ou supprimer un véhicule existant ;
- Afficher la liste des véhicules ;
- Rechercher un véhicule par son code d'immatriculation et affiche l'enregistrement correspondant ;
- Retour au menu principal.

#### **(7) Gestion des parkings**

Ce module permettra de :

- Ajouter un nouveau parking avec toutes ses informations qui vous semblent utiles (N° parking ; capacité ; rue ; arrondissement, ...etc.) ;
- Modifier, ou supprimer un parking existant ;
- Restituer, ou déposer un véhicule dans un parking ;
- Faire sortir un véhicule d'un parking ;
- Afficher la liste de véhicules par parking et le nombre de places vides par parking ;
- Rechercher un parking par son nom et affiche l'enregistrement correspondant ;
- Retour au menu principal.

### **(8) Gestion de sanctions**

Ce module permettra de :

- Régler les sanctions des clients qui ont retardé la restitution des véhicules ;
- Afficher la liste des clients sanctionnés;
- Retour au menu principal.

### **Quelques règles de gestion :**

- Un client peut effectuer une ou plusieurs réservations par appel téléphonique. Ces réservations après validation font l'objet de contrats signés par le client.
- Un contrat ou réservation ne concerne qu'un seul client et un seul véhicule.
- Une réservation qui n'a pas été réglée avant 2 jours de la date de location du véhicule sera annulée.
- Une réservation qui a été réglée avant 2 jours donnera lieu à un contrat.
- Un client peut avoir une ou plusieurs factures.
- Une facture ne concerne qu'un seul client et un seul contrat.
- Un client peut louer une ou plusieurs voitures qui existent déjà dans le parking.
- Un véhicule ne peut être loué que par un seul client et pourra être loué après sa restitution à un autre client.
- Un véhicule peut exister dans un seul parking s'il est restitué.
- Un parking peut contenir plusieurs véhicules.
- Le nombre de véhicules garés dans un parking ne devra pas dépasser la capacité de ce dernier.

### ***Déroulement du mini-projet***

- Ce mini-projet devra être fait **en trinôme(Max) ou binôme(Min)**. Les membres du groupe doivent appartenir à la même filière.
- **Dans la situation où je recevrai des travaux qui représentent un cas de plagiat (copie intégrale ou partielle), la note finale sera divisée par le nombre de groupes qui ont rendu ces travaux identiques.**

### ***Travail à rendre***

- Un court rapport (version papier) d'une longueur comprise entre 10 et 14 pages comportant (à rendre le jour de l'évaluation) :
  - Page de garde
  - Une introduction.

- Partie conception : les diagrammes de cas d'utilisation, diagramme de classe et diagrammes de séquence.
  - l'organisation et la répartition des tâches au sein du groupe durant la durée du projet (brièvement).
  - bilan qualitatif du travail, difficultés rencontrées, points qui vous ont paru intéressants.
  - Une conclusion sur l'apport du projet en termes technique, scientifique, humain.
  - Le code source ne doit pas faire partie du rapport.
- Le code source complet de votre application sous forme d'une archive zip et un jar exécutable doivent être envoyés par courrier électronique à votre professeur avant deux jours de la date de l'évaluation.

### ***Evaluation du mini-projet et calendrier :***

Durée de préparation du mini-projet : **60 jours**. (À compter à partir de **28/03/2022**)

- Le planning de l'évaluation de vos mini-projets vous sera communiqué ultérieurement.
- L'évaluation consiste à consacré 20 min pour chaque groupe.
- L'évaluation sera basée sur les critères suivants :
  - La conception UML de l'application (Diagrammes de cas d'utilisation, Diagramme de classes,...).
  - Qualité technique du code et test de fiabilité de l'application et persistance des données.
  - L'ergonomie des interfaces.
  - Lisibilité du code : présentation du programme, usage de variables ayant des noms explicites, commentaires.
  - Démonstration de l'application et la pertinence des réponses aux questions sur le travail réalisé.
- Le code source complet de votre application sous forme d'une archive zip et un jar exécutable envoyé par mail à votre professeur avant deux jours de la date de l'évaluation.

### **N.B :**

- L'utilisation des bases de données au lieu des fichiers pour la sauvegarde des données de l'application n'est pas obligatoire mais possible.
- Les interfaces de cette application peuvent être créées à l'aide du plugin d'eclipse **WindowBuilder** ou **Scene Builder** l'outil de conception d'interfaces graphique pour JavaFX.