PERIER Hugo
Td B
Année 2017-18

Base de données et interopérabilité

PFR : Agence ESCAPADE

- PLAN -

PERIER Hugo
Td B
Année 2017-18

I- Introduction - Explications générales

Au lancement du programme, le main permet la connexion avec la base de données (« BDD » pour la suite du rapport) via une commande sur MySqlConnection et via les coordonnées du serveur. Puis est appelé : menu(connection)

Cette fonction se charge d'afficher de manière sympathique le menu présentant les 3 fonctionnalités du programme :

- Réservation du séjour
- Check-Out du client
- Tableau de bord de l'agence ESCPADE

Affichage du menu :

Mode d'emploi du programme :

- L'utilisateur doit simplement choisir en entrée le numéro du menu à tester. Puis il n'aura QUE à appuyer sur la toucher « Entrée » du clavier pour faire défiler les différentes étapes.
- A la fin de l'exécution de la fonctionnalité choisie, le programme revient au menu principale.

Important : patientez lors du passage d

PERIER Hugo
Td B
Année 2017-18

II- Explication des codes par étapes

A. Réservation d'un séjour

ETAPE 1 :

La fonction identification_client affiche dans a console l'étape en question et appel successivement plusieurs fonctions.

La première est recuperation_info_fiche_client. : elle renvoi un tableau de string contenant le nom, l'adresse du séjour désiré, la semaine correspondante, et l'arrondissement en fonction des données du rapport cette partie n'est pas pleine de Console.ReadLine() afin de ne pas faire intervenir l'utilisateur et facilité la vérification des fonctionnalités du programme. Cette fonction donne cet affichage dans la console :

```
                              - RESERVATION SEJOUR -


ETAPE n°1    (exemple d'affichage possible, préalablement rempli)
Saisie informations client/sejour

Nom du client ? : Kimzz  [saisie]
Adresse du sejour ? : ESILV, La Defense  [saisie]
Numero de la semaine du séjour désiré ? : 14  [saisie]
Séjour désiré (numero de l'arrondissement) ? : 16  [saisie]

appuyez sur une touche pour continuer
```

Une fois le client créé dans la base de données, un fichier XML **M1** contenant les informations de l'identification client est créé et enregistré dans le dossier « debug » du programme grâce à la fonction creation_XML_info_client. L'explication du code est superflue car identique à la correction des TD5. De même la lecture du fichier se fait avec lecture_XML_info_client.

Affichage correspondant à la création du fichier M1 dans la console :

```
                              - RESERVATION SEJOUR -


ETAPE n°1    Fichier M1
fichier "M1_client_identification.xml" créé dans le fichier Debug.

appuyez sur une touche pour continuer
```

PERIER Hugo
Td B
Année 2017-18

Affichage correspondant à la lecture du fichier M1 dans la console :

Et fichier M1 créé :

```
<?xml version="1.0"?>
- <M1_client_identification>
    <nom>Kimzz</nom>
    <adresse>ESILV, La Defense</adresse>
    <date_sejour>14</date_sejour>
    <sejour>16</sejour>
  </M1_client_identification>
```

PERIER Hugo
Td B
Année 2017-18

ETAPE 2 :

La fonction creation_client_bdd : elle renvoi true si le client a bien pu etre ajouté à la BDD. Elle ouvre la connexion avec le serveur et exécute la requête SQL suivante :

```
"INSERT INTO `ESCAPADE`.`Client` (`idClient`, `nom`, `prenom`, `adresse`, `email`, `num_tel`) VALUES (\"" + new_id + "\",\"" + info_fiche_client[0] + "\",\"" + info_fiche_client[1] + "\",\"" + info_fiche_client [2] + "\",\"" + info_fiche_client[3] + "\",\"" + info_fiche_client[4] + "\") ;";
```

En utilisant les données issues de la fonction identification_client . Puis elle appele la fonction client_est_present_dans_BD qui vérifie si le client identifié par son nom en exécutant la requête SQL suivant :

```
"SELECT count(*) idClient FROM client  WHERE nom=\"" + nom_client + "\";"; // "SELECT Count (*) idClient FROM client WHERE nom=\"" + nom_client + "\";";
```

Si une personne avec ce nom existe déjà dans la base de données on renvois son id en sortie de la fonction client_est_present_dans_BD. Sinon (et s'il existe plus de 1 personne dans la BDD avec le meme nom) cette dernière renvoie un string vide.

De plus, l'idClient à créer (si client non présent dans la BDD) se fait grâce à la fonction generateur_id (génère un string de 11 charactères ici).

Si le client n'est pas présent dans la BDD :

```
                        - RESERVATION SEJOUR -

ETAPE n°2
Recherche présence client dans la base de donnée :

Ce client n'est pas dans la base de données.
Création fiche client :

Vôtre prénom ? : Hugo  [saisie]
Vôtre adresse ? : 1 rue de Saussure  [saisie]
Vôtre adresse mail ? : hugo.perier@devinci.fr  [saisie]
Vôtre numéro de téléphone ? : 0630420209  [saisie]
Reussite ajout client, votre identifiant est : BR0S51KKM9P

appuyez sur une touche pour continuer
```

Si le client est déjà présent dans la BDD :

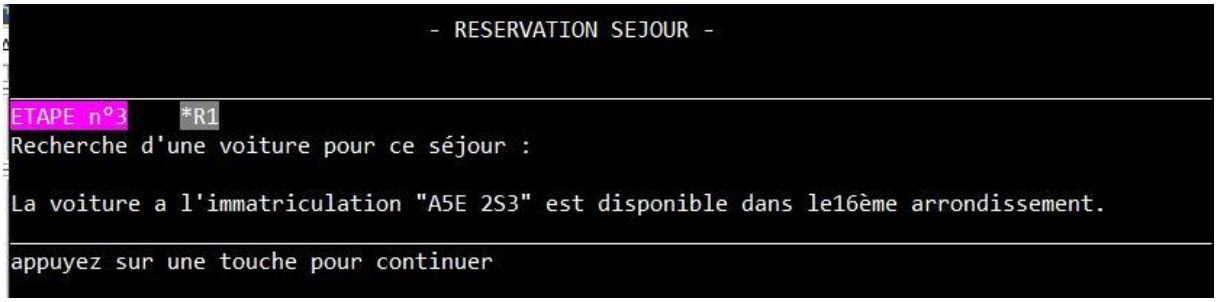
```
                        - RESERVATION SEJOUR -

ETAPE n°2
Recherche présence client dans la base de donnée :

Client présent dans la base de donnée. Identifiant : UNLHAZ7M6RW

appuyez sur une touche pour continuer
```

PERIER Hugo
Td B
Année 2017-18

ETAPE 3 :

La fonction reservation_voiture affiche l'étape dans la console et a besoin de deux autres fonctions. voiture_presente_dans_arrondissement renvoie l'id d'une voiture disponible dans l'arrondissement du séjour désiré (via une simple requête SQL dans la BDD) sinon un string vide. voiture_disponible_ailleurs quant à elle via une simple requête SQL récupère tous les id des voitures disponibles dans paris à cette date de séjour et en récupère le premier.

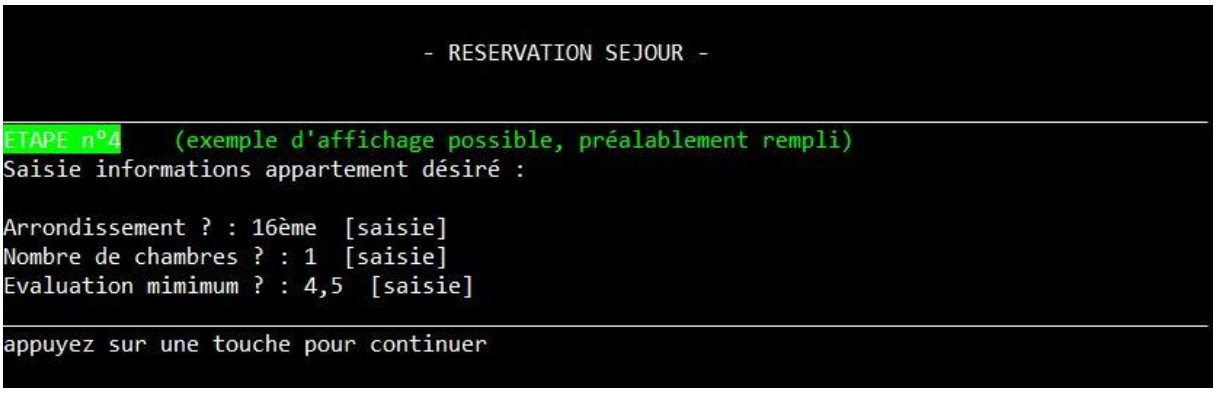
Affichage console de l'étape 3 :



ETAPE 4 :

Simple affichage des possibilités de ce programme si l'on voulait modifier les informations. Ici cette étape se trouve juste un affichage des exigences client, insérées dans un tableau de int via la fonction exigence_client_appartement.

Affichage console de l'étape 4 :



PERIER Hugo
Td B
Année 2017-18

ETAPE 5 :

La fonction selection_appartement renvoi l'appartement à réserver sous type RBNP. Dans un premier temps il a fallu désérialiser le fichier JSON (cf. méthodes et explications du td8 et 9) et créer une liste de RBNP (classe créée au préalable contenant tous les attributs du fichier JSON) grâce a un foreach pour ne sélectionner que les logements possibles. Puis un Console.WriteLite se charge d'afficher le logement dans la console de manière cohérente (grâce a un ToString modifié dans la classe).

Affichage console de l'étape 5 :

```
                              - RESERVATION SEJOUR -


ETAPE n°5     *J3
sélection d'un appartement dans le fichier JSON :

Logement correspondant aux critères du client trouvé :

ID de la chambre : 294489
Quartier : 16
Note de satisfaction : 5
Nombre de chambres : 1
Prix : 174
Disponibilité : yes

appuyez sur une touche pour continuer
```

ETAPE 6 :

La fonction reservation_sejour_mode_non_confirme appele differentes fonctions. Premièrement la fonction creation_sejour_bdd qui renvoie l'id du séjour créé. Pour simplifier l'utilisation du programme le id du séjour crée sera toujours le meme : `"ETKN8U72IDF"`, via une simple requête SQL. Avant de créer ce séjour, il faut déterminer le thème correspondant à l'arrondissement du séjour (connu lui grâce a l'étape n°1). Pour se faire il suffit de lancer une simple requête SQL dans la table 'Theme' . Puis encore pour faciliter l'utilisation du programme, on supprimera le séjour correspondant à la date et au nom du client.

La réussite ou non de l'ajout du séjour sera vérifié grâce a la fonction sejour_est_present_dans_BD vérifiant la présence du séjour en vérifiant le nom et la date (on supposera qu'un client ne réserve pas 2 séjours différents sur le meme week-end).

Enfin un fichier XML M2 de confirmation est créé contenant les differents attributs des réservations.

Affichage console de l'étape 6 :

```
                              - RESERVATION SEJOUR -

ETAPE n°6     Fichier M2
creation du séjour, statut non confirmé

-> Ajout du séjour à la BDD reussit.
-> Fichier "M2_client_confirmation.xml" créé dans le fichier Debug.

appuyez sur une touche pour continuer
```

Et le fichier M2 créé :

```
<?xml version="1.0"?>
- <M2_client_confirmation>
    <numSejour>ETKN8U72IDF</numSejour>
  - <client>
      <nomClient>Kimzz</nomClient>
      <numClient>UNLHAZ7M6RW</numClient>
    </client>
    <nomTheme>KOA</nomTheme>
    <dateSejour>2018-14</dateSejour>
    <statutSejour>sejour valide</statutSejour>
  - <logementReserve>
      <numLogement>294489</numLogement>
      <prixLogement>174</prixLogement>
    </logementReserve>
  - <voitureReservee>
      <idVoiture>A5E 2S3</idVoiture>
    - <caracteristiques>
        <marque>Daihatsu</marque>
        <model>Cabriolet 2 places</model>
      </caracteristiques>
    - <emplacement>
        <parking>Victor Hugo</parking>
        <place>A0</place>
      </emplacement>
    </voitureReservee>
</M2_client_confirmation>
```

PERIER Hugo
Td B
Année 2017-18

ETAPE 7 :

La fonction validation_client appele differentes fonctions :

message_XML_confirmation_client créé le fichier M3 conforme aux exigences du sujet, c'est-à-dire content juste la mention « séjour validé », le numéro du séjour et celui du logement.

validation_s se charge tout simplement de mettre à jour la BDD , de changer le statut du séjour en « valide » et la disponibilité de la voiture.

Affichage console de l'étape 6 :

Et le fichier M3 correspondant :

```
<?xml version="1.0"?>
- <M3_client_validation>
      sejour valide
      <numSejour>ETKN8U72IDF</numSejour>
      <idLogement>294489</idLogement>
  </M3_client_validation>
```

B. Check-Out du client

La fonction update_BDD se charge de mettre à jour la voiture (parking et numéro de place de parking) via les informations recueillis par informations_checkout (encore une fois un simple exemple de ce qui pourrait se faire, mais préalablement rempli dans le but d'une facilitation de l'utilisation du programme).

ATTENTION : afin de faire des UPDATE a ce niveau j'ai été obligé par mon logiciel de faire une manipulation : il faut dans MySQL effectuer :

- Edit- > Preferences->"SQL Editor" et décocher "Safe Update"

Affichage console :

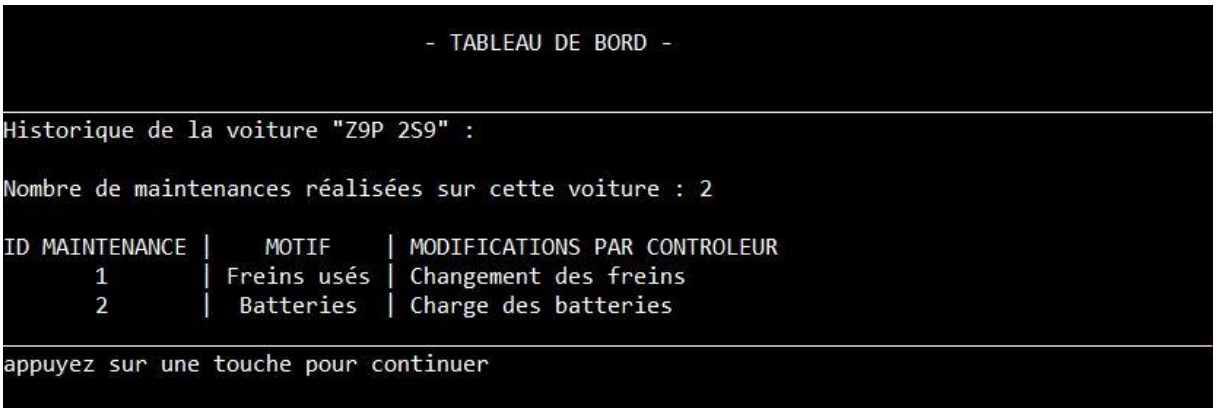
```
                              - CHECK-OUT CLIENT -


ETAPE n°1    (exemple d'affichage possible, préalablement rempli)
Saisie informations du check-out

Numero du séjour ? : ETKN8U72IDF  [saisie]
Arrondissement du parking ou vous avez garé la voiture ? : 21  [saisie]
Numero de la place de parking ou vous l'avez garée ? : A2  [saisie]
Quelle note entre 0-5 donnerez-vous à vôtre séjour ? : 5  [saisie]
Comment avez-vous trouvé ce séjour (en quelques mots) ? : Séjour extraordinaire.  [saisie]

appuyez sur une touche pour continuer
```

C. Tableau de bord

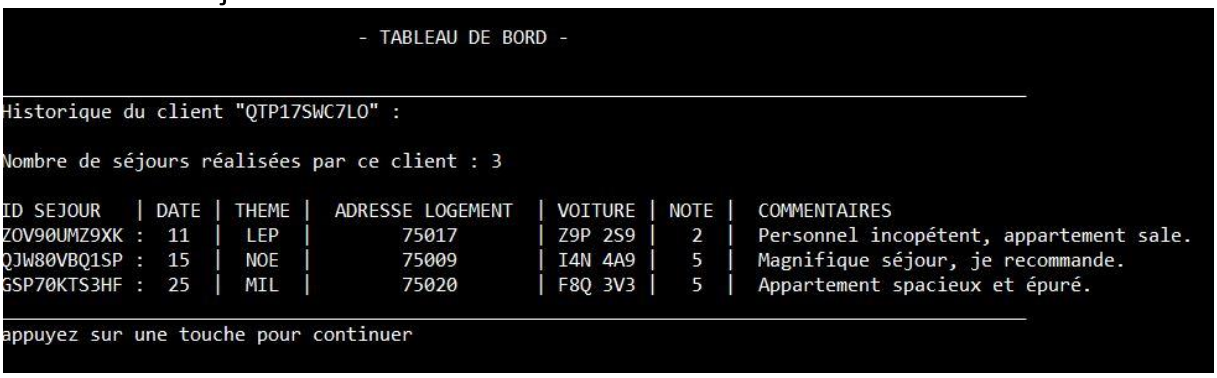
La fonction historique_voitures affiche le nombre de maintenances réalisées pour chaque voiturec(via des requetes SQL issus du rendu n°1).

```
                              - TABLEAU DE BORD -


Historique des voitures (de façon globale):

Nombre de maintenances réalisées pour chaque voiture (idVoiture)
M2J 0H7 : 1 maintenances
P7P 2V3 : 1 maintenances
Z9P 2S9 : 2 maintenances

Nombre de voitures contrôlées par chacun des contrôleurs
Michel Perret s'occupe de 17 voitures.
Luise Bichat s'occupe de 10 voitures.
Bernard Ledore s'occupe de 13 voitures.

appuyez sur une touche pour continuer
```

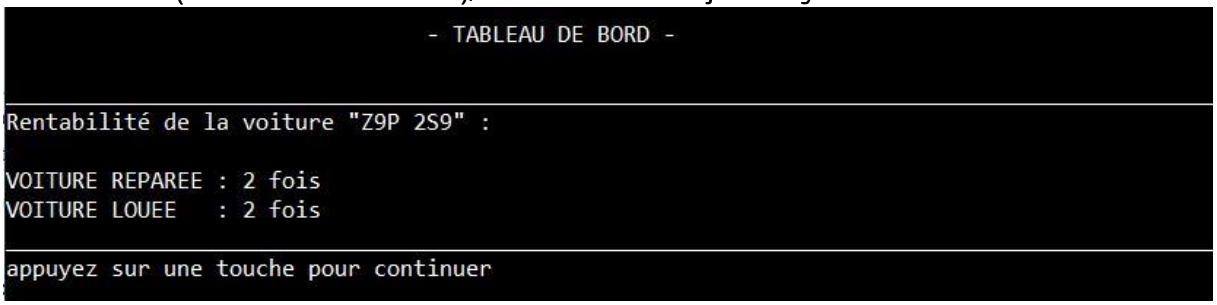
La fonction historique_voiture (sans le 's' ici attention) affiche le nombre de maintenances réalisées sur la voiture identifiée par son id ; et le détail de chaque maintenances).



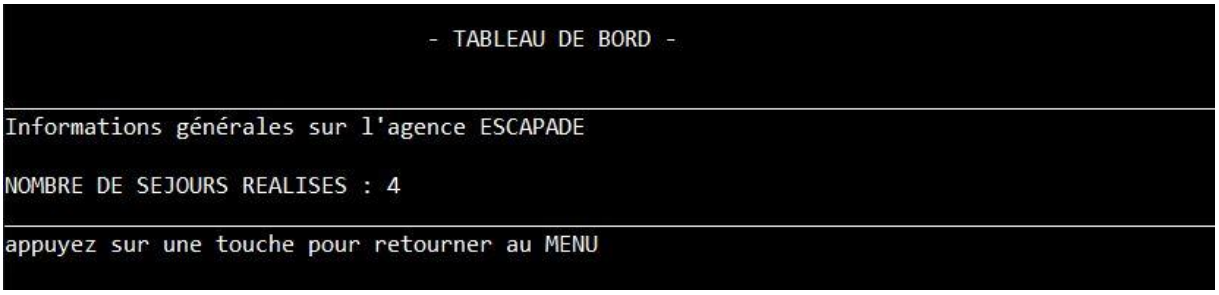
La fonction historique_client affiche le nombre de séjours réalisés par ce client, et le détail de ces séjours.



La fonction rentabilite_voiture affiche le nombre de maintenances ayant eu lieu sur une voiture (identifiée via son id), le nombre de séjours ayant utilisé cette voiture.



Enfin, la fonction information_generale renvoie le nombre de séjours réalisés par l'agence ESCAPADE.



- FIN -