

**Projet #1**  
**Hôtel Plage Santé Inc.**  
**Gestion des réservations et séjours**

Le travail est à réaliser en équipe de deux.

**Date de remise finale : 21 octobre 2024 avant 14h00**

- **Déposer sur LÉA :**

- Le cahier d'analyse – 30 sep. - 10%.
- Les scripts de création de votre base de données – 2 oct. - 20%.
- Le projet en VS – 21 oct. - 65%.
- La gestion des tâches avec Project ou Excel – Chaque semaine - 5%.

**Le cahier d'analyse (10%) : À remettre le 30 septembre 2024 avant 23h55**

Ce document doit être fait avec Word. Prenez soin de la mise en page (page de garde, table des matières, pagination, titre, interligne, ...etc.). Attention aux fautes d'orthographe.

Votre cahier d'analyse doit contenir les éléments suivants :

1. Un résumé descriptif du projet en vos mots. (Une page maximum)
2. Les besoins fonctionnels du système
3. Les acteurs du système et leurs rôles
4. Une planification sommaire : Jalons et tâches approximatives, durées et responsables.
5. Le modèle physique des données (MPD) : Les tables et leurs associations, clés primaires et clés étrangères.

**Les scripts de création de votre base de données (20%) : À remettre le 2 octobre 2024 avant 23h55**

En utilisant SQL Server, créer la base de données et la remplir. Conserver le script (.sql) au propre avec commentaires et le remettre sur Léa.

**Le projet : À remettre le 21 octobre 2024 avant 14h00**

Qualité de l'interface : 5%

Programmation de tous les éléments : 45%

Qualité de votre code : 15%

**La gestion des tâches (5%) : À remettre chaque fin de semaine**

Réalisée avec Project ou Excel, la gestion des tâches doit contenir les jalons et leurs tâches. Pour chaque tâche, vous devez spécifier : le responsable, la durée prévue et la durée réelle ou état d'avancement. (Voir l'exemple fichier Excel).

## LA BASE DE DONNÉES

~~Les numéros des clefs primaires ne doivent pas être automatiques.~~

~~Tous les montants d'argent doivent être de type **NUMERIC(n,2)**.~~

~~Vous pouvez choisir le nombre maximum de chiffres, mais à l'exécution de votre application, lors de la validation d'une donnée, vous devez vous assurer que l'utilisateur n'entre pas une valeur qui dépasse cette limite.~~

~~Toutes les dates doivent être de type **DATETIME**.~~

~~Dans la table **client**, la valeur de *noCli* doit être divisible par 10 (ajoutez une contrainte). Exemple, 10, 20, 30, ...~~

~~Dans la table **invite**, la valeur de *NoInv* ne doit pas être divisible par 10 (ajoutez une contrainte). Exemple, 11, 12, 13, ...~~

~~Le numéro de personne dans la table **PlanifSoin** est soit un no de client, soit un no d'invité.~~

Les numéros des invités d'un client sont définis comme suit :

Le client 10 a comme invités 11, 12, ..., le client 20 a comme invités 21, 22, ... Par conséquent, si dans **planifSoin**, *noPers* vaut 10 on sait qu'il s'agit d'un client et si *noPers* vaut 14, on sait qu'il s'agit d'un invité et que c'est l'invité du client 10.

Ceci limite le nombre d'invités d'un client à 9. Vous allez devoir en tenir compte dans votre application.

~~Dans la table **typeSoin**, il doit y avoir deux (2) types de soins : *Beauté* et *Santé*. Ajoutez ces 2 types de soins.~~

~~Ajoutez une table **typeUtilisateur** qui contient un numéro de type d'utilisateurs et son identification. Il existe deux types d'utilisateurs : *Admin* et *Préposé*~~

~~Ajoutez une table **utilisateur** qui contient des utilisateurs. Cette table doit contenir un numéro d'utilisateur (la clef primaire), un nom d'utilisateur (le nom d'utilisateur est unique), un mot de passe et un numéro de type d'utilisateur.~~

~~Reliez ces deux tables et, dans la table **utilisateur**, insérez au moins un administrateur et au moins un préposé.~~

~~**Vous ne devez pas permettre les suppressions en cascade.** C'est votre application qui doit gérer les suppressions.~~

## NOTES IMPORTANTES SUR LE PROJET

Vous pouvez réaliser ce projet en mode connecté ou en mode déconnecté ou un mélange des deux.

Si vous utilisez les assistants de *Visual Studio* pour construire votre *DataSet*, organisez-vous pour que votre base de données soit bien structurée avant de commencer la programmation du projet sinon vous serez obligés de refaire votre *DataSet* au complet.

~~10% de la note sera attribuée à la qualité de votre interface.~~

Supprimez les boutons de la barre de navigation générée par Visual Studio et programmez vos propres boutons.

Implanter la gestion des conflits d'accès.

Dans le cas d'opérations multiples, utiliser les transactions pour assurer l'intégrité des données de votre base.

Utiliser les outils de validation (listes déroulantes, interception des événements adéquats, MaskedTextBox,...)

Le cas échéant, l'utilisateur doit pouvoir quitter l'application et/ou annuler l'opération en cours.

# LE PROJET

## **1- La connexion au système**

Au démarrage de l'application, l'utilisateur doit s'authentifier à l'aide d'un nom d'utilisateur et d'un mot de passe.

## **2- Le menu**

Si l'authentification réussit, un menu doit s'afficher en fonction du type d'utilisateurs.

**Les administrateurs peuvent :**

1. Gérer les utilisateurs
2. Gérer les clients et leurs invités
3. Gérer les assistants et les soins qu'ils offrent
4. Gérer les soins
5. Planifier des soins pour les clients et leurs invités
6. Gérer les chambres
7. Réserver des chambres (pour les clients)
8. Visualiser des rapports
9. Se déconnecter
10. Quitter l'application

**Les préposés peuvent :**

1. Gérer les clients et leurs invités
2. Planifier des soins pour les clients et leurs invités
3. Réserver des chambres
4. Se déconnecter
5. Quitter l'application

La **déconnexion** permet à l'utilisateur d'authentifier un autre utilisateur.

**Quitter** met fin à l'exécution de l'application.

## **3- Contraintes à respecter**

### **- Contraintes globales**

- Pour toutes les opérations d'ajout, les numéros doivent être générés automatiquement et sont non modifiables.
- Les champs obligatoires doivent être validés à l'exception de ceux qui seront mentionnés plus bas.
- Toutes les opérations de suppression doivent être précédées d'une confirmation de l'utilisateur.
- Après chaque opération, les données doivent être sauvegardées en mémoire et dans la base de données.
- Avant de quitter l'application, sauvegarder dans le format XML (nom\_de\_table.xml), les données des tables *utilisateur*, *client*, *invite*, *assitant*, *assitantSoin*, *soin*, *planificationSoin*, *chambre*, *TypeChambre*, *reservationChambre*

## **~~Gestion des utilisateurs (ajout, suppression et modification)~~**

- **~~Ajout :~~**
  - ~~Tous les champs sont obligatoires.~~
  - **Les noms des utilisateurs doivent être uniques** et non modifiables.
- **~~Modification :~~**
  - **Un utilisateur ne peut modifier son propre type d'utilisateur.** Cependant, il peut modifier le type des autres utilisateurs.
- **~~Suppression :~~**
  - **Un utilisateur ne peut supprimer son propre compte, mais peut supprimer d'autres utilisateurs.** On ne devrait jamais supprimer l'administrateur initial (exple no 1).

## **Gestion des clients et leurs invités (par l'administrateur ou le préposé)**

Le formulaire principal doit afficher tous les clients ainsi que les invités du client sélectionné.

### **A. Gestion des clients**

- **~~Ajout :~~**
  - ~~Tous les champs sont obligatoires~~
  - ~~La date du jour (en lecture seule) est la date d'inscription~~
  - ~~Il n'est pas nécessaire de valider le code postal (varie d'un pays à un autre).~~
- **~~Modification :~~**
  - ~~Hormis le numéro de client et sa date d'inscription, tous les champs peuvent être modifiés.~~
- **~~Suppression :~~**
  - **On ne peut supprimer un client dans les cas suivants :**
    - **S'il a des invités**
    - **S'il a réservé une chambre**
    - **Si des soins ont été planifiés pour lui**

### **B. Gestion des invités**

- **~~Ajout :~~**
  - ~~Respecter les règles de numérotation ainsi que le nombre maximum d'invités par client~~
  - ~~S'il y a un trou, ce dernier doit être comblé. Par exemple, s'il y avait 3 invités qui portaient respectivement les numéros 11, 12 et 13 et qu'on a supprimé l'invité 11, le numéro du prochain invité sera 11.~~
- **~~Modification :~~**
  - ~~Seul le nom est modifiable~~

- **Suppression :**

- On ne peut supprimer un invité si des soins ont été planifiés pour lui.

## - Gestion des assistants et leurs soins (par l'administrateur)

Le formulaire principal doit afficher tous les assistants ainsi que les soins offerts par l'assistant sélectionné

### A. Gestion des assistants :

- **Ajout :**

- Seul le champ remarques est facultatif.

- **Modification :**

- Hormis le numéro d'assistant, tous les champs sont modifiables.

- **Suppression :**

- On ne peut supprimer un assistant s'il a des planifications de soins lui sont assignées.
- La suppression d'un assistant implique la suppression de tous les soins qu'il offre.

### B. Gestion des soins :

- **Ajout :**

- On peut ajouter de nouveaux soins à un assistant

- **Suppression :**

- On ne peut supprimer un soin offert par un assistant si le soin est présent dans une planification.

## ~~- Gestion des soins (par l'administrateur)~~

- ~~○ **Ajout :**~~

- ~~▪ Tous les champs sont obligatoires.~~
- ~~▪ Pour simplifier les calculs, tous les soins durent 60 minutes~~

- ~~○ **Modification :**~~

- ~~▪ Hormis le numéro et la durée du soin, tous les champs sont modifiables.~~

- ~~○ **Suppression :**~~

- ~~▪ On ne peut supprimer un soin offert par un assistant ou planifié.~~

## - Planification des soins (par l'administrateur et le proposé)

- **Ajout d'une réservation soin :**

- Pour éviter les erreurs, le client ou l'invité, le soin et l'assistant seront choisis à partir de listes déroulantes :

- Liste des invités ou clients (nom et numéro)
  - Liste des soins (numéro et description)
  - Liste des assistants (numéro, prénom et nom) qui offrent le soin sélectionné.
- Utiliser le contrôle *DateTimePicker* pour le choix de la date de réservation
  - L'heure de réservation est entre 8h et 17h.
  - La date de réservation ne doit pas être antérieure à la date du jour.
  - La date de réservation ne doit pas être un samedi ou un dimanche
  - Un assistant ne peut donner un soin à la même date/heure à deux personnes.
- **Suppression d'une réservation de soin :**
    - On ne peut supprimer un soin déjà offert.

## - Gestion des chambres et des types de chambres (par l'administrateur)

### ~~A. Gestion des chambres~~

- ~~Ajout~~
  - ~~Tous les champs sont obligatoires~~
- ~~Modification :~~
  - ~~Hormis le numéro, tous les champs sont modifiables.~~
- **Suppression :**
  - On ne peut supprimer une chambre réservée.

### ~~B. Gestion des types de chambres~~

~~Le formulaire principal doit afficher tous les types de chambres ainsi que les chambres (numéro, emplacement, décoration) du type sélectionné.~~

- ~~Ajout :~~
  - ~~La description du type doit être unique~~
- ~~Modification :~~
  - ~~Tous les champs sont modifiables (sauf le numéro)~~
- **Suppression :**
  - On ne peut supprimer un type de chambre, s'il existe des chambres de ce type.

## - Réservation des chambres (par l'administrateur ou le préposé)

- **Ajout :**
  - Pour éviter les erreurs, il vous est recommandé d'utiliser les contrôles suivants :
    - Le nom et le numéro du client à partir d'une liste



- La chambre (numéro, Emplacement, Décoration, Prix saison haute, prix saison basse, prix saison moyenne) à partir d'un *DataGridView*
  - La date du début et de fin du séjour à partir d'un *DateTimePicker*
  - Le nombre de personnes à partir d'une liste ou un *NuméricUpDown* (entre 1 et 6 inclusivement)
  - Toutes les vérifications nécessaires de dates doivent être effectuées
  - S'assurer que la chambre est bien disponible pour toute la période désirée.
- **Suppression**
    - On ne peut annuler une réservation si la date de début du séjour est antérieure à la date actuelle.

## **- La visualisation des rapports (par l'administrateur)**

### **A. Rapport des soins offerts aux clients et aux invités**

L'administrateur doit pouvoir sélectionner un client (numéro et nom complet) ou un invité (numéro et nom complet) et afficher la liste (*DataGridView*) des soins (description, date et heure, prix et nom complet de l'assistant) offerts à la personne choisie. De plus, ce rapport doit afficher le montant total des soins.

### **B. Rapport des réservations de chambres**

L'administrateur doit pouvoir sélectionner une chambre (numéro et emplacement) et afficher l'historique. Une ligne contient le nom complet du client (qui a occupé ou qui occupera la chambre), les dates d'arrivée et de départ ainsi que le nombre de personnes.

### **C. Rapport des soins journaliers des assistants**

L'administrateur doit pouvoir sélectionner un assistant (numéro et nom complet) et une date et afficher la liste des soins offerts par cet assistant à la date choisie. La sélection d'un soin permet de visualiser la liste (*DataGridView*) des personnes qui ont reçu ou recevront le soin choisi à la date donnée. Une ligne du *DataGridView* contient la description, la date et l'heure ainsi que le nom de la personne ayant bénéficié du soin.