



COMPTE RENDU ATELIER PRO

LEPERS Léon & VIEILLARD--GRELET Tom



Application C# de gestion de la base de données d'un hôtel.

07 NOVEMBRE 2023

BTS SIO

Lycée Lamartine

Table des matières

I. Expression des besoins	2
1. Contexte, domaine, processus métier	2
2. Demandeur, acteurs, utilisateurs	2
3. Description de la demande, objectifs, bénéfice attendu & spécifications fonctionnelles	2
4. Contraintes ou exigences.	2
II. Conception, Spécifications techniques	3
1. Description de la solution	3
2. Outils logiciels de la solution	3
3. Architecture matérielle et logicielle de la solution	3
4. Besoins techniques, ressources (humaines, matérielles, logicielles et budgétaires, coûts...)	5
5. Analyse des données (modélisation, diagramme de classes, schéma relationnel)	5
6. IHM (Interfaces homme-machine), Maquettage	6
7. Conduite de projet : décomposition en tâches, structure des équipes, planning, durée	7
III. Développement	8
1. Réalisation des interfaces et programmes conformes aux spécifications fonctionnelles attendues	8
2. Dossier de programmation, codes sources documentés et commentés	10
IV. Exploitation, mise en production	12
1. Tests	12
2. Bilan	13

I. Expression des besoins

1. Contexte, domaine, processus métier

Les hôtels Balladins nous demandent la réalisation d'une application pour faciliter la gestion de leurs hôtels. Hôtels Balladins est une entreprise à but lucratif avec une activité touristique : la gestion d'hôtels.

2. Demandeur, acteurs, utilisateurs

Les demandeurs sont le service informatique de Balladins qui s'occupent du développement des différents hôtels. Ils ont recours aux services de deux développeurs freelances, messieurs LEPERS Léon et VIEILLARD--GRELET Tom, qui se chargeront de la réalisation du projet durant l'automne 2023. L'application a pour objectif d'être utilisée par les différents gérants des hôtels Balladins partout en France, afin de faciliter leur gestion des hôtels..

3. Description de la demande, objectifs, bénéfice attendu & spécifications fonctionnelles

L'application demandée est un logiciel de gestion possédant plusieurs fonctionnalités. Le logiciel a accès à la base de données des hôtels qui comporte hôtels, chambres, réservations, équipements... Le logiciel doit répondre à un cahier des charges qui demande plusieurs formulaires de gestion des différentes informations.

Comme indiqué précédemment, ce logiciel vise à faciliter la gestion des hôtels pour le gérant ou toute personne travaillant dans ce domaine. Les avantages attendus par Balladins pour ce logiciel sont une économie de temps et d'énergie pour les employés impliqués, leur permettant ainsi de consacrer plus de ressources à d'autres problèmes ou projets.

4. Contraintes ou exigences.

Le logiciel doit être réalisé grâce aux outils Visual Studio et SQL Server, afin de lier programmation objet et base de données. Le projet est entièrement réalisé sur les machines personnelles des deux développeurs, qui sont donc plus familiers avec leurs environnements de travail, et donc plus aptes à réussir le projet. Les délais étant d'environ 2 mois, le logiciel doit être rendu le 8 novembre 2023 dernière heure, sous peine de pénalité de salaire pour chaque jour de retard. La gratification des développeurs sera attribuée en points supplémentaires sur leur bilan semestriel.

II. Conception, Spécifications techniques

1. Description de la solution

La solution proposée par les développeurs est un logiciel de gestion, possédant plusieurs formulaires, tous répondant au cahier des charges demandé par Balladins. Il y a donc 4 formulaires de gestion différents, ainsi qu'un formulaire d'accueil au lancement du logiciel. Le cahier des charges est composé de 4 grandes idées, premièrement le développement d'un outil de connexion au logiciel pour que le gérant d'un hôtel puisse accéder à la gestion de celui-ci. Ensuite, la création d'un formulaire de gestion des informations de l'hôtel afin de gérer et pouvoir modifier ou supprimer les informations relatives à l'hôtel. Puis, la réalisation d'un formulaire de gestion des chambres afin de pouvoir ajouter ou supprimer des chambres. Ensuite le développement d'un formulaire de gestion des réservations, qui permet de créer une réservation en sélectionnant une chambre et une date. On peut aussi voir les différentes réservations d'une date choisie, ainsi que les chambres disponibles ou non dans le cas de la création d'une réservation.

2. Outils logiciels de la solution

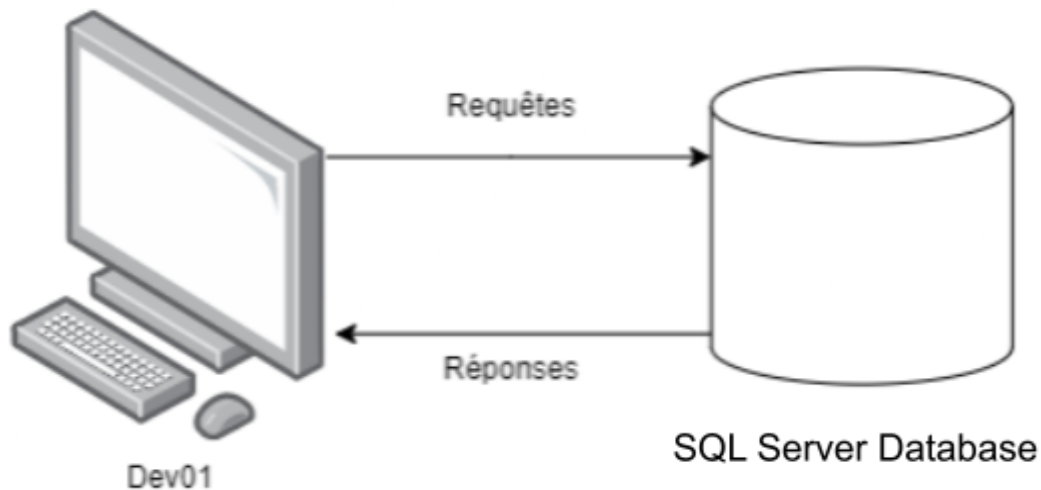
Le logiciel est intégralement développé sous Windows 10, sur le logiciel Visual Studio en langage C#, en utilisant l'outil WinForms pour créer les formulaires. Les développeurs utilisent Entity Framework pour faire le lien entre Visual Studio et la base de données SQL Server, qui contient un jeu d'essai d'hôtels.

3. Architecture matérielle et logicielle de la solution



Mise en commun du projet

Pour travailler plus efficacement, les développeurs utilisent l'outil de dépôt GitHub qui leur permet de mettre en commun et de partager les avancées sur le projet virtuellement.



Interaction application/base de données

L'application est en relation avec une base de données (celle de Pathé Mâcon). Cette dernière reçoit des requêtes SQL de l'application et renvoie des réponses (exemple : liste des projections diffusées X jour) cherchées dans la base de données.



Architecture du logiciel

Comme décrit précédemment, le logiciel possède un formulaire d'accueil au lancement du projet, qui possède un menu conduisant vers les trois autres formulaires de gestion.

4. Besoins techniques, ressources (humaines, matérielles, logicielles et budgétaires, coûts...)

Le projet nécessite plusieurs types de ressources, premièrement les ressources humaines, comme précisé, le travail est réalisé par un binôme de développeurs freelances. Les besoins matériels sont pris en charge par les développeurs étant donné qu'ils travaillent l'un comme l'autre sur leurs machines personnelles. Les besoins logiciels sont un peu plus conséquents, le principal logiciel utilisé est Visual Studio pour gérer toute la programmation objet en langage C# et les formulaires, mais il y a aussi SQL Server et EntityFramework pour la gestion de la base de données. Il n'y a pas de coûts supplémentaires.

5. Analyse des données (modélisation, diagramme de classes, schéma relationnel)

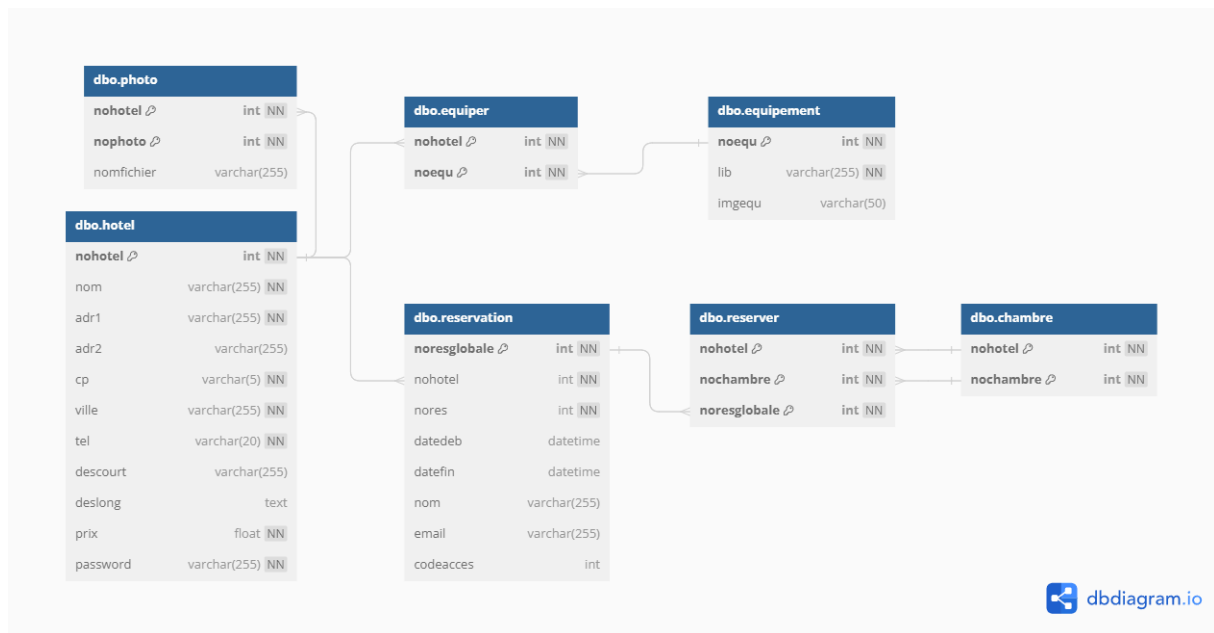
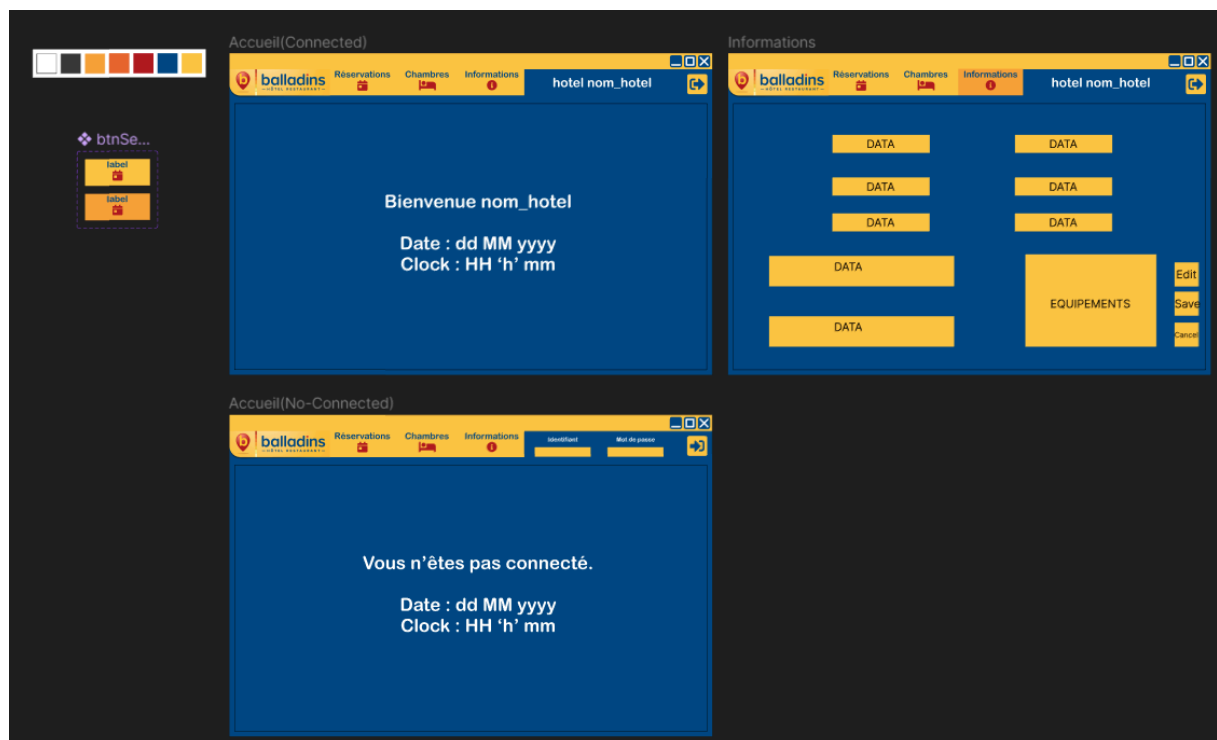
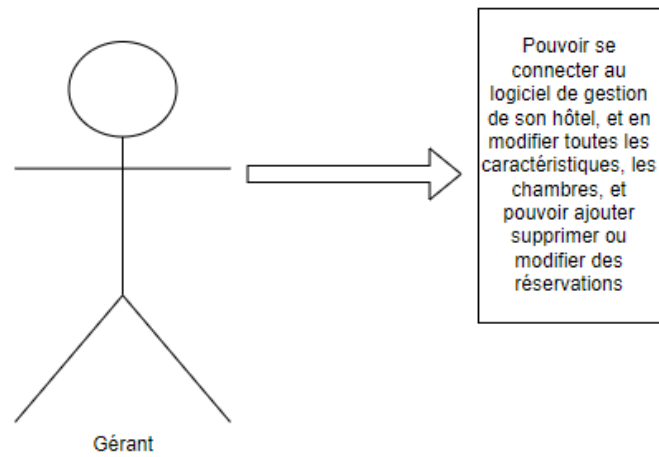


Schéma relationnel de la base de données

6. IHM (Interfaces homme-machine), Maquettage



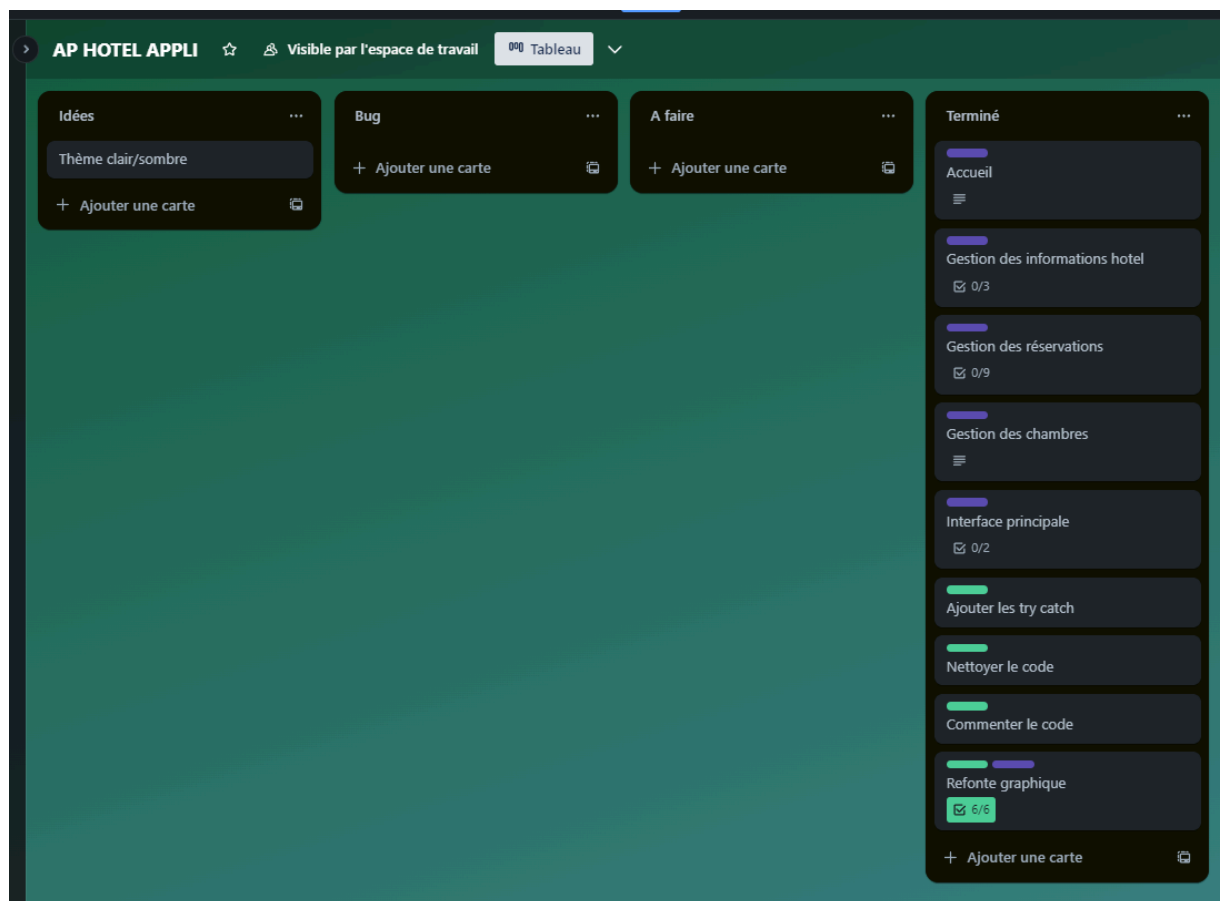
7. Conduite de projet : décomposition en tâches, structure des équipes, planning, durée

Les tâches à réaliser sont, comme précisé plus tôt, un minimum de 4 formulaires à réaliser (voir [description de la solution](#)). Les développeurs se sont partagés les tâches de la façon suivante :

- Formulaires **Informations, Réservations** → VIEILLARD--GRELET Tom
- Formulaires **Chambre et Connexion** → LEPERS Léon

Chacun a développé sa partie du projet qu'ils ont mis en commun sur GitHub. Pour ce qui est du planning, les délais ont été respectés, et le projet a été fini dans les temps. Le strict minimum pour les formulaires a été complété environ 1 mois avant la fin des délais, ce qui a donné le temps aux développeurs pour apporter des améliorations, optimisations dans le code et interface graphique.

TRELLO utilisé pour la gestion du projet :



III. Développement

1. Réalisation des interfaces et programmes conformes aux spécifications fonctionnelles attendues

Lancement du logiciel, accès au formulaire d'accueil, besoin de se connecter afin de pouvoir accéder aux formulaires de gestion.

Formulaire d'accueil :



Accès aux différents formulaires possible depuis la barre de navigation, et affichage de l'heure et de la date du jour, ainsi que d'une information quant à l'état de connexion de l'utilisateur.

Formulaire de gestion des informations (Première fonctionnalité requise dans le cahier des charges) :



Présence de champs et de listes pour afficher toutes les informations, ainsi que d'un bouton pour accéder au mode édition de ces informations.

Formulaire de gestion des chambres (Deuxième fonctionnalité requise dans le cahier des charges) :

Outil de gestion Hotel Balladins

balladins - HÔTEL RESTAURANT -

Réservations Chambres Informations

Hotel LYON - DARDILLY

Gestion des chambres

N°1	N°2	N°3	N°4	N°5	N°6	N°7	N°8	N°9	N°10	N°11	N°12	N°13	N°14	N°15	N°16
N°17	N°18	N°19	N°20	N°21	N°22	N°23	N°24	N°25	N°26	N°27	N°28	N°29	N°30	N°31	N°32
N°33	N°34	N°35	N°36	N°37	N°38	N°39	N°40	N°41	N°42	N°43	N°44	N°45	N°46	N°47	N°48
N°49	N°50	N°51	N°52	N°53	N°54	N°55	N°56	N°57	N°58	N°59	N°60	N°61	N°62	N°63	N°64
N°65															

66

+

🗑️

Génération de boutons pour chaque chambre de l'hôtel, ainsi que de boutons pour ajouter ou supprimer une/des chambres, et d'un outil pour attribuer un numéro de chambre personnalisé.

Formulaire de gestion des réservations (troisième fonctionnalité demandée dans le cahier des charges) :

Premier formulaire (création de réservation) :

Outil de gestion Hotel Balladins

balladins - HÔTEL RESTAURANT -

Réservations Chambres Informations

Hotel LYON - DARDILLY

Gestion des réservations

• Création de réservation • Visualisation des réservations

Début réservation

lundi 06 novembre 2023 à 16h46

Fin réservation :

lundi 06 novembre 2023 à 16h46

Nom :

Mail :

+ Ajouter la réservation

Chambres disponibles :

<input type="checkbox"/> n°1	<input type="checkbox"/> n°22	<input type="checkbox"/> n°43	<input type="checkbox"/> n°64
<input type="checkbox"/> n°2	<input type="checkbox"/> n°23	<input type="checkbox"/> n°44	<input type="checkbox"/> n°65
<input type="checkbox"/> n°3	<input type="checkbox"/> n°24	<input type="checkbox"/> n°45	
<input type="checkbox"/> n°4	<input type="checkbox"/> n°25	<input type="checkbox"/> n°46	
<input type="checkbox"/> n°5	<input type="checkbox"/> n°26	<input type="checkbox"/> n°47	
<input type="checkbox"/> n°6	<input type="checkbox"/> n°27	<input type="checkbox"/> n°48	
<input type="checkbox"/> n°7	<input type="checkbox"/> n°28	<input type="checkbox"/> n°49	
<input type="checkbox"/> n°8	<input type="checkbox"/> n°29	<input type="checkbox"/> n°50	
<input type="checkbox"/> n°9	<input type="checkbox"/> n°30	<input type="checkbox"/> n°51	
<input type="checkbox"/> n°10	<input type="checkbox"/> n°31	<input type="checkbox"/> n°52	
<input type="checkbox"/> n°11	<input type="checkbox"/> n°32	<input type="checkbox"/> n°53	
<input type="checkbox"/> n°12	<input type="checkbox"/> n°33	<input type="checkbox"/> n°54	
<input type="checkbox"/> n°13	<input type="checkbox"/> n°34	<input type="checkbox"/> n°55	
<input type="checkbox"/> n°14	<input type="checkbox"/> n°35	<input type="checkbox"/> n°56	
<input type="checkbox"/> n°15	<input type="checkbox"/> n°36	<input type="checkbox"/> n°57	
<input type="checkbox"/> n°16	<input type="checkbox"/> n°37	<input type="checkbox"/> n°58	
<input type="checkbox"/> n°17	<input type="checkbox"/> n°38	<input type="checkbox"/> n°59	
<input type="checkbox"/> n°18	<input type="checkbox"/> n°39	<input type="checkbox"/> n°60	
<input type="checkbox"/> n°19	<input type="checkbox"/> n°40	<input type="checkbox"/> n°61	
<input type="checkbox"/> n°20	<input type="checkbox"/> n°41	<input type="checkbox"/> n°62	
<input type="checkbox"/> n°21	<input type="checkbox"/> n°42	<input type="checkbox"/> n°63	

Choix d'une date de début et d'une date de fin, ainsi que d'un nom et d'un mail, puis des chambres afin de créer une réservation et l'ajouter à la base de données.

Deuxième formulaire (visualisation des réservations) :

Outil de gestion Hotel Balladins

balladins — HÔTEL RESTAURANT —

Réservations **Chambres** **Informations**

Hotel LYON - DARDILLY

Gestion des réservations

● **Création de réservation** ● **Visualisation des réservations**

Voir les réservation du **mardi 7 novembre 2023**

Les réservations :

n°1

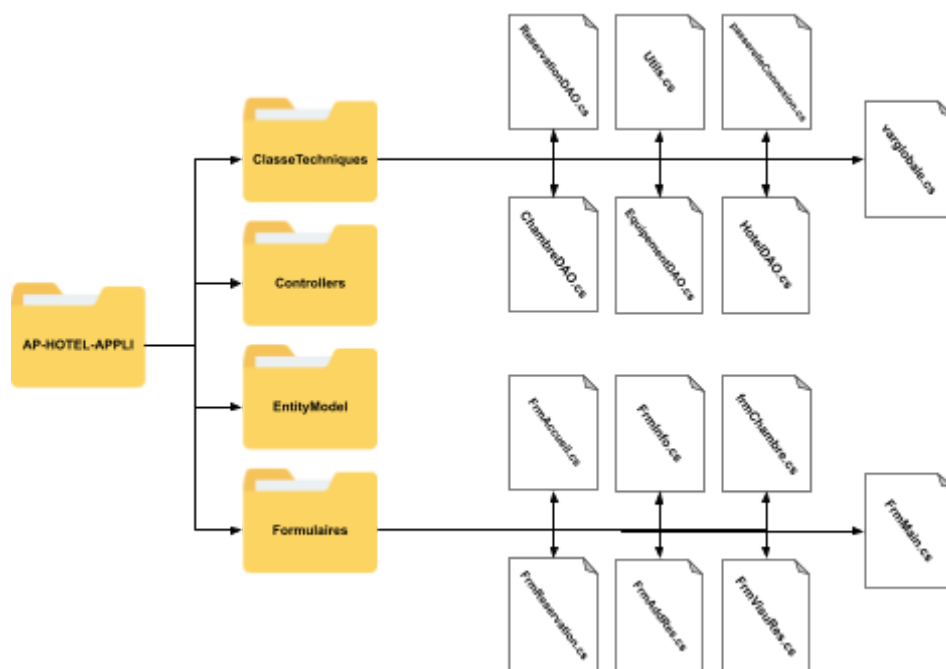
Résumé de la réservation

N°1
 Nom : jean
 Email : jean@mail.com
 Début : mardi 7 novembre 19h00
 Fin : jeudi 9 novembre 08h00
 Code d'accès : 52753

<input type="checkbox"/> n°1	<input type="checkbox"/> n°17	<input type="checkbox"/> n°33	<input type="checkbox"/> n°49	<input type="checkbox"/> n°65
<input type="checkbox"/> n°2	<input type="checkbox"/> n°18	<input type="checkbox"/> n°34	<input type="checkbox"/> n°50	
<input type="checkbox"/> n°3	<input type="checkbox"/> n°19	<input type="checkbox"/> n°35	<input type="checkbox"/> n°51	
<input type="checkbox"/> n°4	<input type="checkbox"/> n°20	<input type="checkbox"/> n°36	<input type="checkbox"/> n°52	
<input checked="" type="checkbox"/> n°5	<input type="checkbox"/> n°21	<input type="checkbox"/> n°37	<input type="checkbox"/> n°53	
<input type="checkbox"/> n°6	<input type="checkbox"/> n°22	<input type="checkbox"/> n°38	<input type="checkbox"/> n°54	
<input type="checkbox"/> n°7	<input type="checkbox"/> n°23	<input type="checkbox"/> n°39	<input type="checkbox"/> n°55	
<input type="checkbox"/> n°8	<input type="checkbox"/> n°24	<input type="checkbox"/> n°40	<input type="checkbox"/> n°56	
<input type="checkbox"/> n°9	<input type="checkbox"/> n°25	<input type="checkbox"/> n°41	<input type="checkbox"/> n°57	
<input type="checkbox"/> n°10	<input type="checkbox"/> n°26	<input type="checkbox"/> n°42	<input type="checkbox"/> n°58	
<input type="checkbox"/> n°11	<input checked="" type="checkbox"/> n°27	<input type="checkbox"/> n°43	<input type="checkbox"/> n°59	
<input type="checkbox"/> n°12	<input type="checkbox"/> n°28	<input type="checkbox"/> n°44	<input type="checkbox"/> n°60	
<input type="checkbox"/> n°13	<input type="checkbox"/> n°29	<input type="checkbox"/> n°45	<input type="checkbox"/> n°61	
<input type="checkbox"/> n°14	<input type="checkbox"/> n°30	<input type="checkbox"/> n°46	<input type="checkbox"/> n°62	
<input type="checkbox"/> n°15	<input type="checkbox"/> n°31	<input type="checkbox"/> n°47	<input type="checkbox"/> n°63	
<input type="checkbox"/> n°16	<input type="checkbox"/> n°32	<input type="checkbox"/> n°48	<input type="checkbox"/> n°64	

Choix d'une date puis visualisation des réservations prévues pour cette date, des différentes informations relatives à la réservation. Possibilité de modifier ou de supprimer une réservation.

2. Dossier de programmation, codes sources documentés et commentés



Arborescence du projet

Extraits de code du projet

Méthode permettant de récupérer la liste des chambres disponibles de l'hôtel en fonction des dates sélectionnées et/ou de la réservation sélectionnée :

```
/// <summary>
/// Retourne la liste des chambres disponible de l'hôtel pour les dates [dateTimeDebut, dateTimeFin] et/ou la réservation lareservation
/// </summary>
/// <param name="dateTimeDebut"></param>
/// <param name="dateTimeFin"></param>
/// <param name="lareservation">[optionnel]</param>
/// <returns>Liste de chambre</returns>
public static List<chambre> GetLesChambresDisponibles(List<chambre> lesChambresHotel, DateTime dateTimeDebut, DateTime dateTimeFin, reservation lareservation = null)
{
    try {
        if (lareservation == null)
        {
            // Filtrer les chambres qui n'ont pas de réservation qui chevauche la période [dateTimeDebut, dateTimeFin]
            return lesChambresHotel.Where(chambre =>
                !chambre.reservation.Any(reservation =>
                    !(dateTimeDebut >= reservation.datefin || dateTimeFin <= reservation.datedeb)
                )
            ).OrderBy(c => c.nochambre).ToList();
        }
        else
        {
            // Filtrer les chambres qui n'ont pas de réservation qui chevauche la période [lareservation.datedeb, lareservation.datefin]
            // ou qui font partie de la réservation lareservation
            return lesChambresHotel.Where(chambre =>
                chambre.reservation.Contains(lareservation) ||
                !chambre.reservation.Any(reservation =>
                    (reservation.datedeb <= lareservation.datefin && reservation.datefin >= lareservation.datedeb)
                )
            ).OrderBy(c => c.nochambre).ToList();
        }
    }
    catch (Exception ex)
    {
        Utils.GenerateFileError(ex);
        return null;
    }
}
```

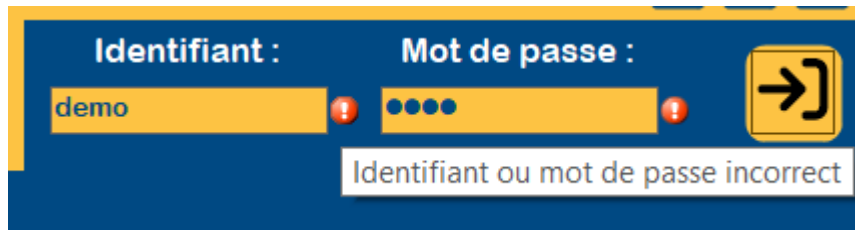
Méthode permettant d'initialiser la connexion à un hôtel en l'instanciant dans une variable globale:

```
/// <summary>
/// Permet d'initialiser la connexion à un hôtel en fonction de l'identifiant et du mot de passe.
/// </summary>
/// <param name="id">Identifiant d'hôtel</param>
/// <param name="password">Mot de passe d'hôtel</param>
/// <returns>true, si un hôtel correspond à l'identifiant et au mot de passe. Sinon, false</returns>
public static bool InitConnexion(string id, string password)
{
    try
    {
        // Vérifie si l'identifiant et le mot de passe saisis par l'utilisateur correspondent à un hôtel
        if (varglobale.connexion.hotel.Where(hotel => hotel.nom == id && hotel.password == password).Any())
        {
            // Initialise la connexion à l'hôtel correspondant à l'identifiant et au mot de passe
            varglobale.hotel = varglobale.connexion.hotel.Where(hotel => hotel.nom == id && hotel.password == password).FirstOrDefault();
            return true;
        }
        else return false;
    }
    catch (Exception ex)
    {
        Utils.GenerateFileError(ex);
        return false;
    }
}
```

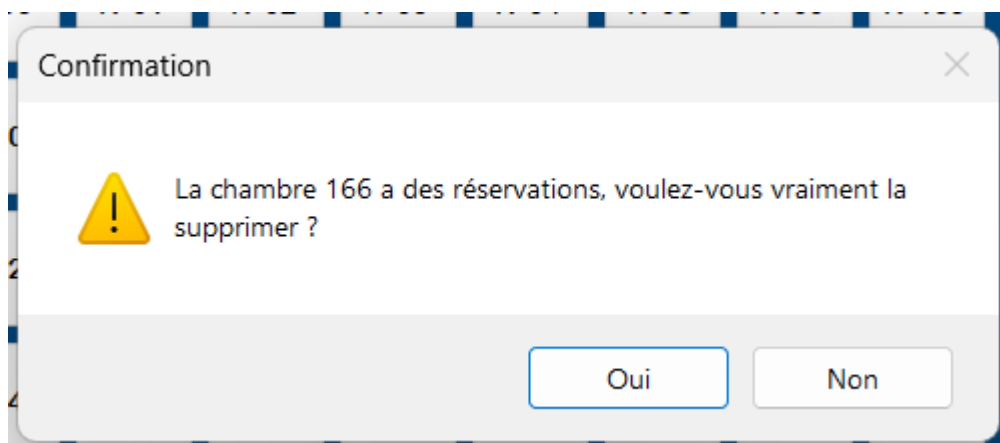
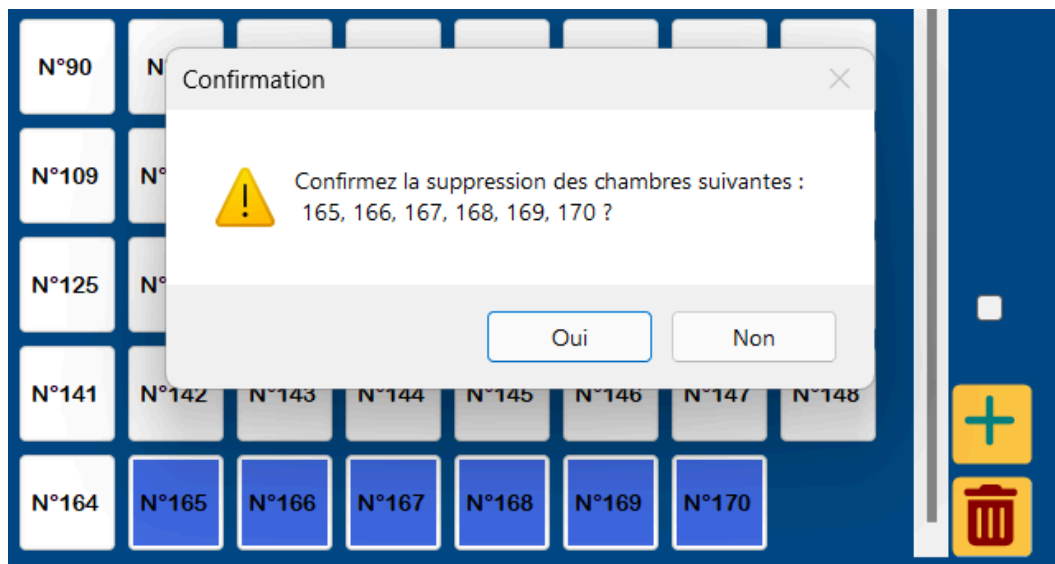
IV. Exploitation, mise en production

1. Tests

Les tests ont été réalisés avec le jeu de données fourni. Présence d'icônes qui préviennent lors d'une erreur pendant le remplissage d'un champ (ou d'un oubli), et présence de tests dans le code pour éviter la moindre erreur, et dans le cas où une erreur apparaîtrait, un fichier logs est créé et accessible.



Exemple de “warnings” qui apparaissent lors d’une erreur à la connexion -> ici pour un identifiant ou un mot de passe mal saisi.



Lors de la suppression d’une chambre, si celle-ci possède une ou des réservations, message de confirmation afin d’éviter les éventuelles erreurs.

2. Bilan

Les deux développeurs ont terminé la réalisation du projet à temps, sans dépasser les délais, et toutes les fonctionnalités demandées sont fonctionnelles, voire pour certaines plus optimisées que ce qui était demandé au départ. Ce qui en ressort pour les deux développeurs, c'est surtout de l'expérience, de nouvelles méthodes de travail et une familiarisation avec de nouveaux outils, tels que Entity Framework, qui pourront surement leur être utiles pour leurs futurs projets. Leur principal objectif était de viser une utilisation simple et fluide du logiciel, le tout sur un design moderne.