

Dossier de projet

Hôtel du Rivage

Application de réception d'hôtellerie et site internet d'hôtel



HOTEL
DU RIVAGE

Auteur	Maëlle Huguenin-Dezot et Alexandre Pitarch
Date de début de projet	13.06.24
Date de fin de projet	02.07.24
Date de reddition du rapport	18.06.24
Modification	02.07.24
Version	rapport-de-projet-GRP3-hotel-MHD-API

Table des matières

Table des matières	2
1 Résumé à l'intention de la direction	4
2 Initialisation	5
2.1 Description du projet	5
2.2 Objectifs	5
2.3 Contraintes.....	5
2.3.1 Contraintes techniques :.....	5
2.3.2 Contraintes organisationnelles :.....	6
2.3.3 Contraintes des délais :.....	6
2.4 Organisation du travail.....	6
2.4.1 Microsoft Teams.....	6
2.4.2 GitHub.....	6
2.5 Planification du projet.....	6
2.5.1 Diagramme de Gantt prévisionnel.....	7
2.5.2 Diagramme de Gantt final	9
3 Conception	11
3.1 Liste et brève description des fonctionnalités.....	11
3.2 Architecture du système.....	11
3.3 Base de données	12
3.3.1 Dictionnaire de données.....	12
3.3.2 Modèle conceptuel de données (MCD).....	12
3.3.3 Modèle logique de données (MLD)	14
3.3.4 Ajout des Noms de Tables au Pluriel et Préfixe "tb_"	15
3.4 Maquettes.....	15
4 Réalisation	19
4.1 MySQL WorkBench	19
4.1.1 ERD.....	19
4.1.2 Détail du code particulier	20
4.1.3 Protocoles de tests MySQL.....	20
4.2 Gestion des clients	23
4.2.1 Nouveau client.....	23
4.2.2 Détail du code particulier	23
4.2.3 Protocoles de tests	23

4.3	Gestion des réservations.....	24
4.3.1	Nouvelle réservation	24
4.3.2	Détail du code particulier	25
4.3.3	Protocoles de tests	25
4.4	Gestion des chambres.....	26
4.4.1	Ajout d'une chambre	26
4.4.2	Détail du code particulier	26
4.4.3	Protocoles de tests	27
4.5	Site Web	27
4.5.1	Fonctionnalités Principales.....	28
4.5.2	Fonctionnalités Clés.....	28
4.5.3	Technologies Utilisées	28
4.5.4	Formulaire de contact	28
4.5.5	Détail du code particulier	29
4.5.6	Protocoles de tests	29
4.5.7	QR code.....	29
5	Livraison.....	30
5.1	Mode d'emploi utilisateur	30
6	Conclusions.....	31
6.1	Conclusion Maëlle.....	31
6.2	Conclusion Alexandre	31
7	Annexes	32
7.1	Code source	32
7.2	Source	32
7.3	Table des illustrations	32

1 Résumé à l'intention de la direction

À l'ESIG dans le cadre de l'atelier C822, du 13.06.2024 au 02.07.2024, Maëlle et Alexandre avons développé une application de gestion des réservations d'hôtel ainsi qu'un site web correspondant. L'application, réalisée en C# avec une interface WPF, permet aux réceptionnistes de gérer en temps réel les chambres, clients, localités et réservations via une base de données MySQL hébergée sur Infomaniak. Notre objectif était de mettre en avant les compétences acquises durant cette année scolaire.

Le site web est responsive, développé en HTML5, CSS3, Bootstrap et JavaScript, et offre une interface conviviale pour que les clients puissent contacter l'hôtel via un formulaire sécurisé. Chaque envoi de formulaire de contact met à jour notre base de données MySQL.

Pour l'organisation du projet, nous avons utilisé Microsoft Teams et GitHub pour le développement, structurant notre code en différentes branches pour éviter les conflits. La planification a été suivie à l'aide d'un diagramme de Gantt créé via GanttProject, assurant le respect des délais et des dépendances entre les tâches. Nous avons tous les deux contribué à toutes les étapes du projet, ce qui nous a permis de développer des compétences communes et de nous entraider.

MySQL Workbench a été notre outil pour concevoir la base de données, à partir d'un modèle conceptuel de données (MCD) transformé en modèle logique de données (MLD), puis en schéma ERD. Nous avons utilisé le forward engineering pour créer les tables et les relations, incluant un trigger pour automatiser l'enregistrement des dates de réservation. Des tests ont été effectués pour valider la fiabilité de la base de données, couvrant toutes les opérations essentielles telles que l'insertion, la modification et la suppression des données.

Nous avons rencontré des difficultés avec la suppression et la modification des chambres et étages en raison de l'utilisation de clés naturelles et des dépendances entre les entités, ce qui a rendu ces opérations complexes à gérer. La table associative pour les réservations a également posé des défis pour rechercher les informations des clés étrangères dans le repository, mais nous avons réussi malgré tout. Nous aurions souhaité ajouter des fonctionnalités supplémentaires, comme une table d'archive de réservations ou l'affichage en rouge des réservations non payées, mais le temps nous a manqué.

À la suite de cela, nous avons créé une plateforme répondant efficacement aux besoins spécifiques d'un environnement hôtelier, en tenant compte des exigences particulières de notre hôtel fictif, l'Hôtel du Rivage. Pour un déploiement en conditions réelles, nous envisagerions d'ajouter des fonctionnalités supplémentaires afin d'améliorer encore l'efficacité et l'utilité de notre solution.

Ce projet nous a permis de développer de nouvelles compétences essentielles, telles que l'intégration d'une base de données à une application et le debugging, cruciales dans un environnement professionnel. La collaboration avec GitHub a renforcé notre capacité à travailler en équipe et à gérer efficacement le code. La création d'une application complète, depuis le site web jusqu'à la base de données, a enrichi notre compréhension globale du développement logiciel.

En conséquence, ce projet a consolidé nos compétences techniques et renforcé notre capacité à travailler en équipe. Nous avons acquis de l'expérience en intégration de bases de données, développement d'applications WPF et création de sites web responsive. Les défis rencontrés nous ont préparés à aborder de futurs projets avec plus de confiance et d'efficacité.

2 Initialisation

2.1 Description du projet

Le but de ce projet est de démontrer les compétences acquises au cours de l'année scolaire.

Après avoir reçu les informations sur le projet, nous avons décidé de développer une application de gestion des réservations d'hôtel et un site web associé. L'application serait destinée aux réceptionnistes d'un hôtel pour gérer les chambres, les clients, les localités et les réservations. De plus, le site web permettra aux clients de s'inscrire et d'enregistrer leurs informations directement dans la base de données centrale.

Le but de ce travail scolaire est de créer une application et un site web connectés via une base de données sur un serveur Infomaniak. Les tables de la base de données sont créées avec MySQL Workbench, après avoir élaboré un dictionnaire de données, un MCD, un MLD et un ERD pour assurer une conception structurée. Un QR code a également été développé pour accéder directement au site web de l'hôtel. Enfin, l'application permet l'impression des réservations des chambres.

2.2 Objectifs

L'objectif est de centraliser et de synchroniser les données entre l'application et le site web, facilitant ainsi une gestion efficace et en temps réel des réservations. Cette intégration peut permettre à plusieurs employés de travailler simultanément avec l'application, améliorant la productivité et réduisant les erreurs. De plus, les réceptionnistes auront la possibilité d'imprimer les détails des réservations directement depuis l'application, assurant une gestion pratique et rapide des opérations quotidiennes.

2.3 Contraintes

Le projet doit respecter plusieurs contraintes techniques, organisationnelles et de collaboration :

2.3.1 Contraintes techniques :

- **Base de données :**
Créer un dictionnaire de données, un Modèle Conceptuel de Données (MCD), un Modèle Logique de Données (MLD) et un diagramme ERD (Entity-Relationship Diagram). Les tables doivent être créées avec MySQL Workbench et connectées à un serveur Infomaniak.
- **Application :**
Développer une application WPF en C# avec Visual Studio. L'application doit être responsive et inclure des contrôles de champs en C#. Elle doit également permettre l'impression des réservations.
- **Site web :**
Développer un site web en HTML, JavaScript et CSS (nous avons décidé la framework Bootstrap, à la place de CSS, ce qui n'était pas obligatoire). Le site doit être responsive et inclure des contrôles de champs en JavaScript. Il doit être mis en ligne à l'adresse suivante : <https://lmo.divtec.me/2023/pi02-grpe3>.
- **QR Code :**
Générer un QR code permettant d'accéder directement au site web de l'hôtel.
- **Formulaires de gestion :**
Développer au moins deux formulaires de gestion basés sur les tables principales (CRUD) et un formulaire de gestion basé sur deux tables avec une relation maître-détail (formulaire –

sous-formulaire). Inclure des fonctionnalités de filtres, de tri, de recherche de données, d'édition et d'impression d'états.

2.3.2 Contraintes organisationnelles :

- **Planification :**
Créer un diagramme de Gantt pour planifier les différentes phases et étapes du projet. Le projet doit être terminé pour le 2 juillet 2024 à 15h00, incluant la documentation (rapport de projet et tutoriel utilisateur), les exécutable et les sources.
- **Collaboration :**
Établir un espace de travail sur GitHub et un dossier partagé via Microsoft Teams pour faciliter la collaboration entre les membres de l'équipe. Chaque participant doit contribuer au développement en C#, au développement web et à la documentation.
- **Documentation :**
Documenter le projet en rédigeant un rapport de projet complet et en créant un tutoriel utilisateur. Des maquettes et des protocoles de tests doivent également être élaborés.

2.3.3 Contraintes des délais :

- Le projet doit être achevé et livré avec toute la documentation, les exécutable et les sources pour le **2 juillet 2024 à 15h00**.
- La défense du projet est prévue pour le **4 juillet 2024**.

Ces contraintes doivent être respectées pour assurer le bon déroulement du projet et répondre à toutes les exigences, garantissant ainsi son achèvement réussi.

2.4 Organisation du travail

2.4.1 Microsoft Teams

Pour l'organisation du travail, nous avons créé un groupe Teams où nous avons structuré différents dossiers pour partager nos sources et documents.

2.4.2 GitHub

Nous avons utilisé GitHub pour coder et créer différentes branches selon les fonctionnalités sur lesquelles nous travaillons, afin d'éviter les conflits de code. Nous avons organisé notre dépôt en deux dossiers principaux : "Web" pour tout ce qui concerne le site internet et "Windows" pour l'application.

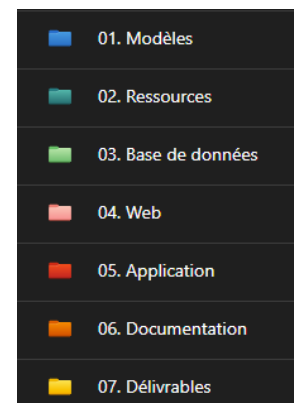


Figure 1, arborescence dossier Microsoft Teams

2.5 Planification du projet

Pour planifier notre projet, nous avons décidé de faire un diagramme de Gantt que nous avons créé grâce à l'application GanttProject. Celle-ci nous permet de visualiser le temps que nous avons pour effectuer chaque tâche et la dépendance qu'il y a entre elles. De plus nous savons clairement où sont les deadlines (jalons).

2.5.1 Diagramme de Gantt prévisionnel

Voici notre diagramme de Gantt

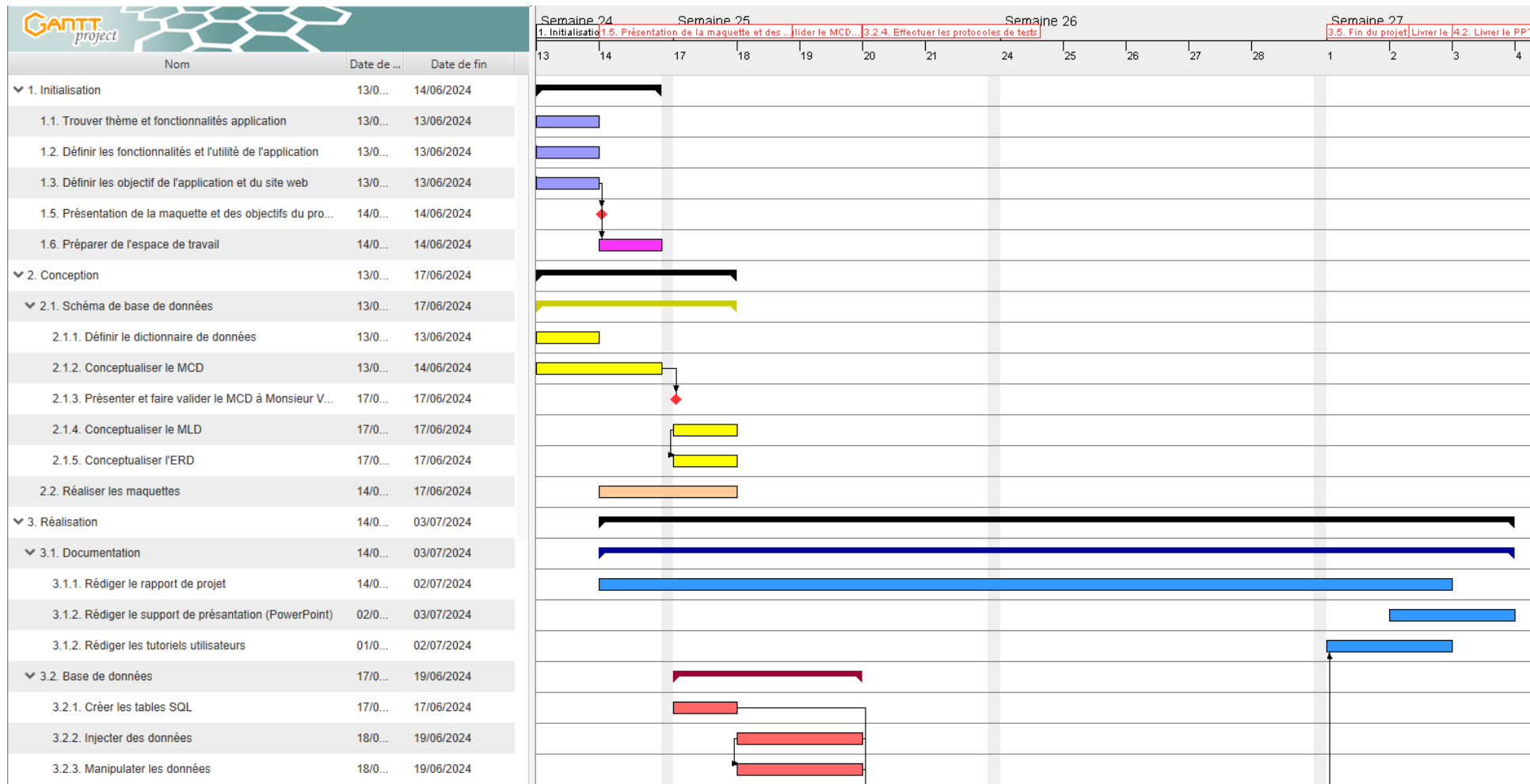


Figure 2, Gantt prévisionnel partie 1

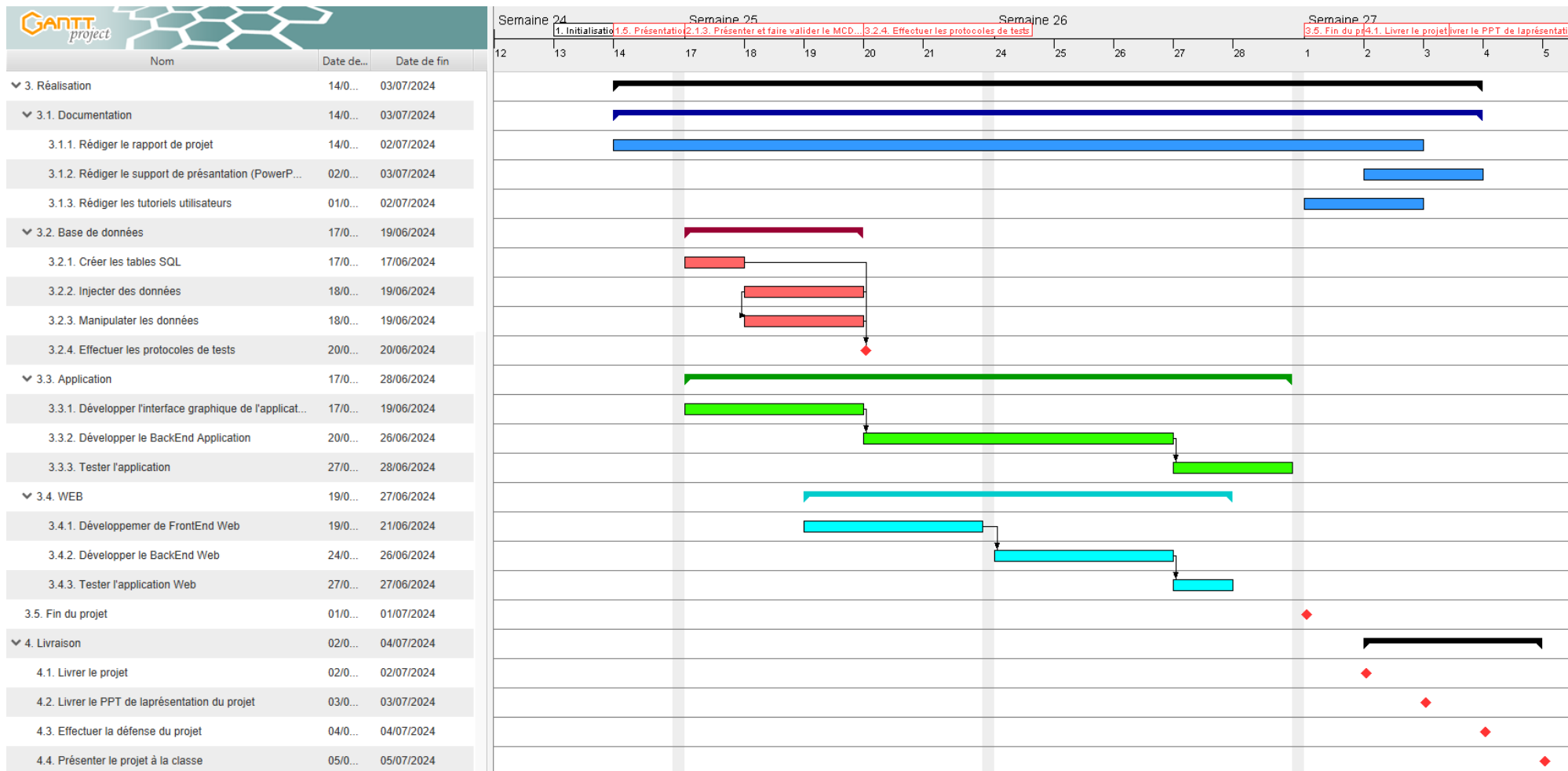


Figure 3, Gantt prévisionnel partie 2

2.5.2 Diagramme de Gantt final

La phase d'initialisation et de conception s'est déroulée comme prévu, tandis que la phase de réalisation a pris plus de temps, en particulier pour le développement de l'application.

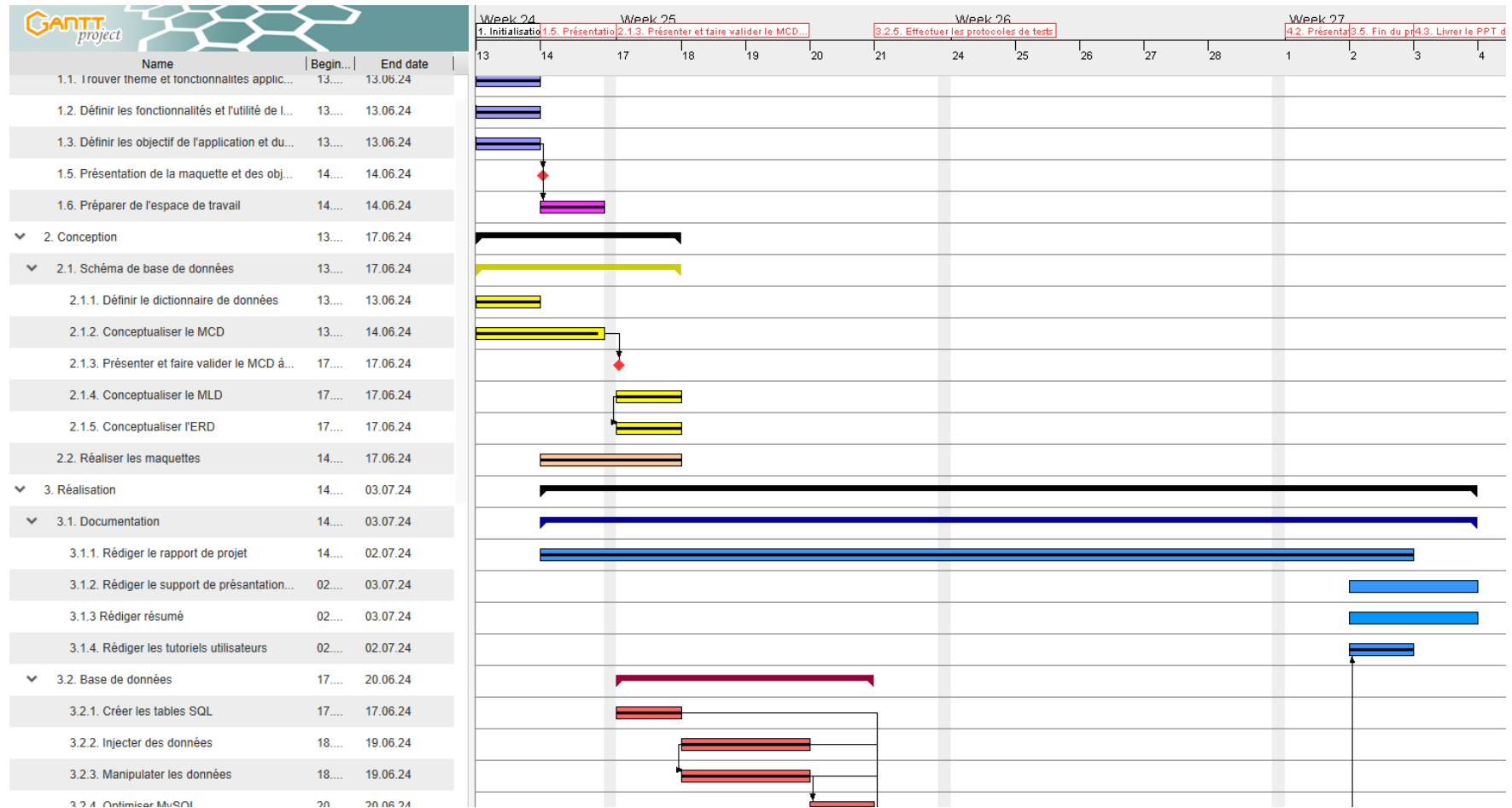


Figure 4, Gantt final partie 1

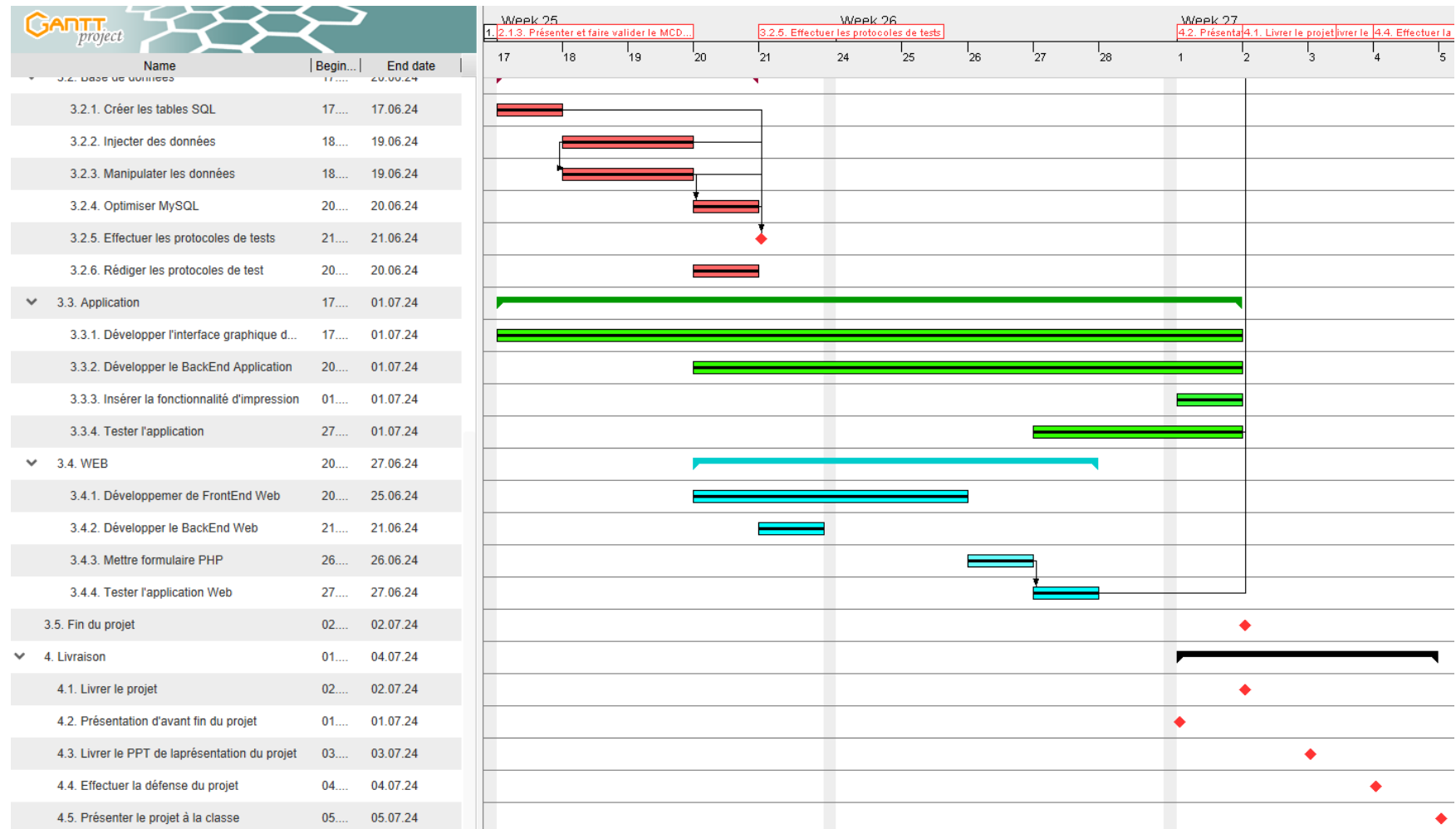


Figure 5, Gantt final partie 2

3 Conception

3.1 Liste et brève description des fonctionnalités

Notre projet consiste en une application de gestion des réservations de chambres d'hôtel et un site web d'inscription des clients.

Les principales fonctionnalités de l'application incluent la gestion des chambres (ajout, modification, suppression), des clients (enregistrement, mise à jour, suppression), des localités (ajout, modification, suppression) et des réservations (création, modification, annulation). Le site web, quant à lui, présente l'hôtel et permet aux clients de s'inscrire en ligne, synchronisant automatiquement leurs données avec la base de données utilisée par l'application de gestion. Bien que les clients ne puissent pas effectuer de réservations via le site web, cela montre notre capacité à lier une application de bureau et un site web à une base de données commune, assurant la mise à jour en temps réel des informations.

3.2 Architecture du système

Cette architecture gère un système de réservation d'hôtel en ligne. Les clients accèdent au site web via leurs appareils personnels et remplissent un formulaire de contact. Ce formulaire est envoyé via Internet à une base de données MySQL hébergée sur un hébergeur Infomaniak.

Les réceptionnistes utilisent une application C# sur leur ordinateur pour accéder aux informations de la base de données. L'application permet de gérer les réservations en insérant les détails de la chambre et les dates de réservation.

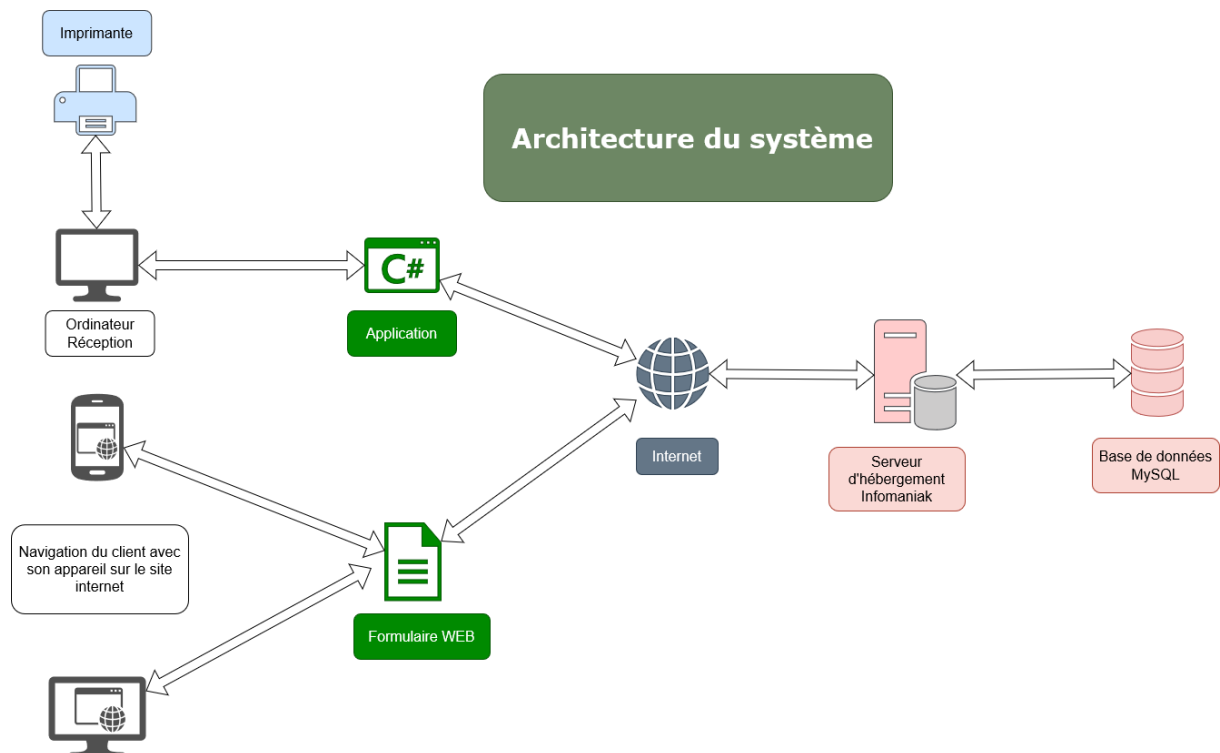


Figure 6, architecture du système de réservation

Une imprimante est intégrée pour imprimer les confirmations de réservation et d'autres documents nécessaires. Cette architecture assure une gestion centralisée et efficace des réservations, améliorant la réactivité et la qualité du service client.

3.3 Base de données

3.3.1 Dictionnaire de données

Suite à la décision de créer une application pour gérer les réservations des chambres d'un hôtel, nous avons élaboré un dictionnaire de données avant de commencer le Modèle Conceptuel de Données (MCD). Cela nous a permis de structurer notre approche et de nous poser les bonnes questions pour avoir une conception cohérente et efficace de la base de données.

Le dictionnaire de données comprend les éléments suivants :

Nom symbolique	Conteneur	Domaine ou type	Règles de calcul
Nom du village/ville	Localité	token	Mandatory
NPA	Localité	word	Mandatory
Nom	Client	token	Mandatory
Prénom	Client	token	Mandatory
Rue	Client	token	Mandatory
Téléphone	Client	token	Mandatory
Adresse email	Client	token	Mandatory
Date de naissance	Client	date	Mandatory / minimum 18 ans
Numéro de réservation	Réservation	int (AID)	Mandatory
Nombre de personne	Réservation	int	Mandatory / self.nbrPers <= Chambre.nbrLits
Déjeuner	Réservation	bool	Mandatory
Date réservation client	Réservation	date	Automatique mandatory
Date arrivée	Réservation	date	Mandatory
Date départ	Réservation	date	Mandatory
Statut Libre/Occupé	Chambre	bool	Mandatory
Nombre de lits	Chambre	int	Mandatory
Numéro de chambre	Chambre	word(NID)	Mandatory
Type de chambre	Type de chambre	token	Mandatory
Numéro etage	Etage	word	Mandatory / pas plus petit que 1

3.3.2 Modèle conceptuel de données (MCD)

3.3.2.1 Explication :

Le Modèle Conceptuel de Données (MCD) de notre système de réservation d'hôtel en ligne est structuré autour des entités clés suivantes : Client, Localité, Réservation, Chambre et Étage. Voici les principales règles de gestion et les relations entre ces entités :

- Client
 - Un client utilise le système pour effectuer des réservations.
 - Chaque client est associé à une localité unique.
 - Un client peut effectuer plusieurs réservations.
- Localité
 - Représente le lieu de résidence d'un client.
 - Chaque localité peut être associée à plusieurs clients.
- Réservation
 - Représente une demande de réservation de chambre par un client.
 - Chaque réservation est effectuée par un seul client.
 - Chaque réservation est associée à une seule chambre.
- Chambre
 - Représente une unité d'hébergement dans l'hôtel.
 - Une chambre peut être associée à plusieurs réservations.
 - Chaque chambre est située à un seul étage.
- Étage
 - Représente un niveau dans l'hôtel où sont situées les chambres.
 - Chaque étage possède un identifiant unique (NID) pour indiquer son numéro.
 - Chaque étage peut comporter plusieurs chambres.

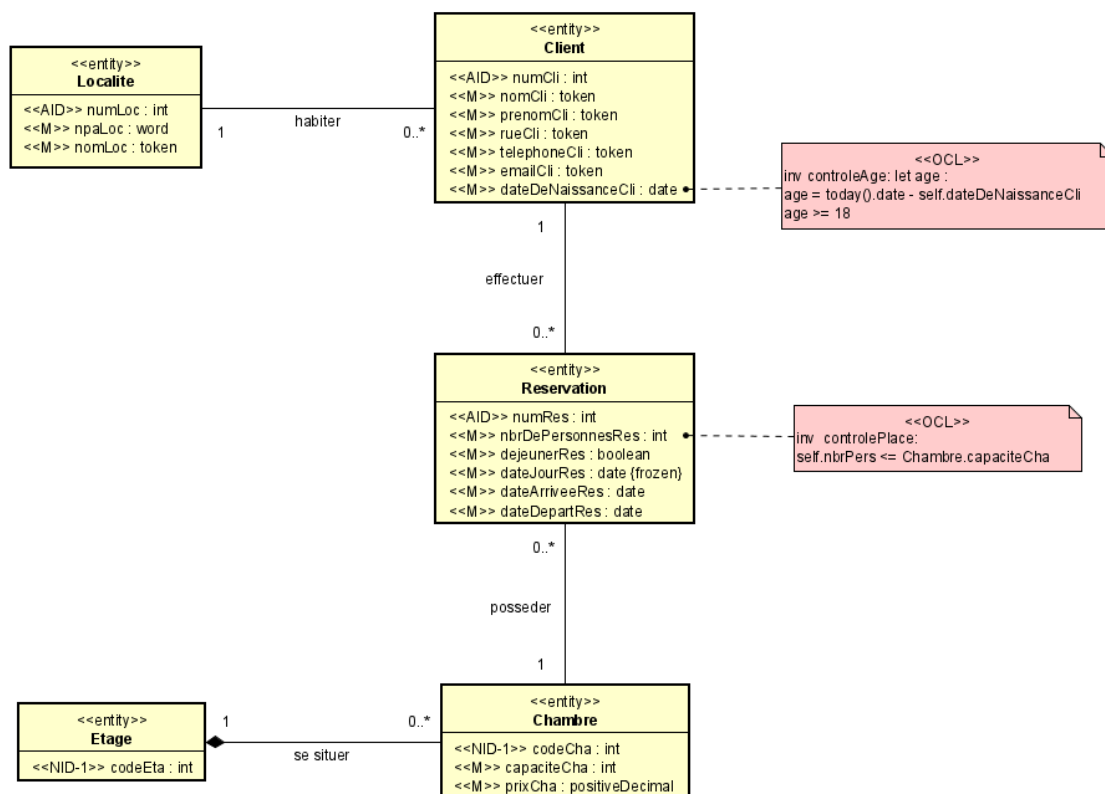


Figure 7, modèle conceptuel de données

3.3.2.2 Contraintes OCL pour le Modèle Conceptuel de Données (MCD)

Pour garantir l'intégrité et la validation des données dans notre système de réservation d'hôtel en ligne, nous utilisons le langage de contrainte Object Constraint Language (OCL). Voici deux contraintes spécifiques définies pour notre modèle.

Ces contraintes OCL sont intégrées dans le modèle conceptuel de données pour garantir la conformité et la fiabilité des opérations de réservation dans notre système d'hôtel en ligne.

3.3.2.3 *Contrôle d'âge pour la réservation :*

Contrôle d'âge pour la réservation :

inv controleAge:

```
let age: Integer = Date::today().year - self.client.dateDeNaissance.year in  
age >= 18
```

Cette contrainte vérifie que l'utilisateur effectuant une réservation est âgé d'au moins 18 ans, en calculant l'âge à partir de la date de naissance du client.

Nous voulions l'insérer dans la base de données, mais nous avons fini par la mettre de les contrôles de saisi de notre application et site web.

3.3.2.4 *Contrôle de capacité de la chambre pour la réservation :*

inv controlePlace:

```
self.nbrPers <= self.chambre.capaciteCha
```

Cette contrainte s'assure que le nombre de personnes incluses dans une réservation n'excède pas la capacité maximale de la chambre réservée.

3.3.3 *Modèle logique de données (MLD)*

Le modèle logique de données (MLD) est une représentation structurée des données après avoir ajouté les clés primaires (PK), les clés étrangères (FK) et les ponts de clé étrangère (PFK), ainsi que d'autres éléments nécessaires à la gestion des données dans la base de données. Le MLD nous a aider à créer notre ERD (point 4.1.1).

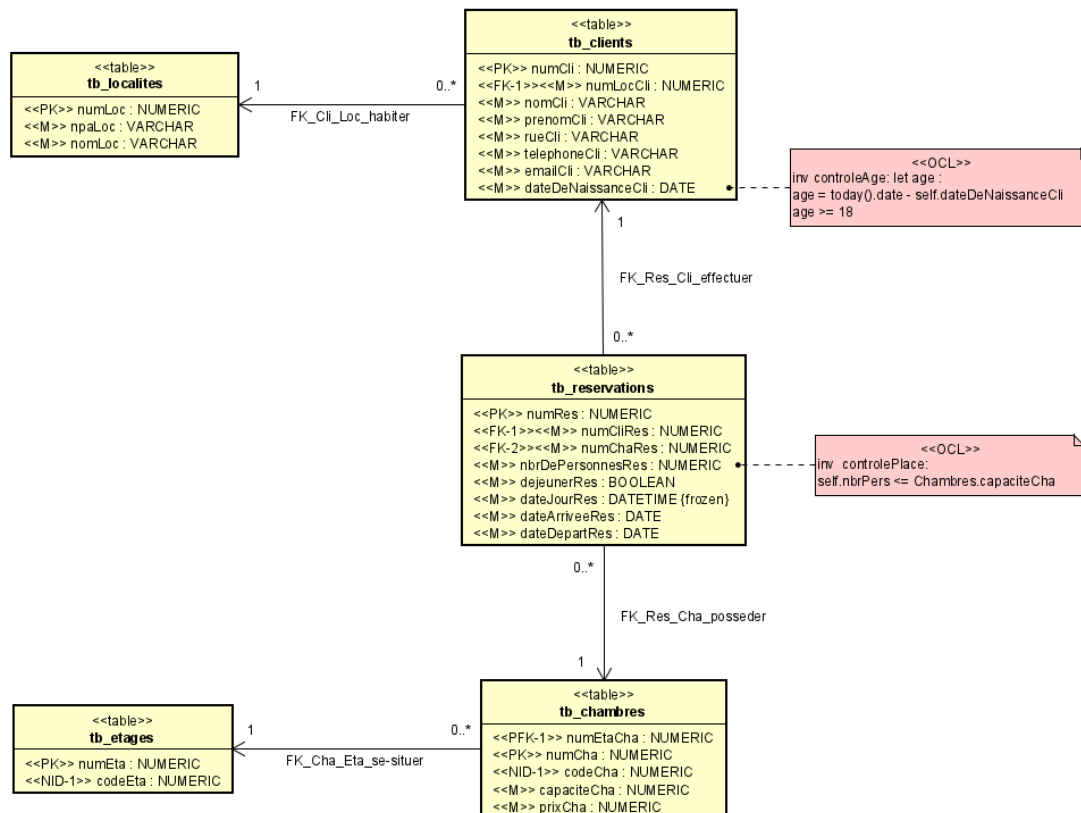


Figure 8, modèle logique de données

3.3.4 Ajout des Noms de Tables au Pluriel et Préfixe "tb_"

Pour améliorer la clarté et la cohérence du modèle, chaque nom de table est désormais écrit au pluriel et préfixé par "tb_", ce qui facilite l'identification des tables dans le schéma de la base de données.

3.4 Maquettes

Nous avons réalisé des maquettes avec Figma pour déterminer le design de notre application et de notre formulaire web. Nous avons décidé que les couleurs principales de notre projet seraient le bleu, le blanc et le gris.

Voici le lien de notre espace de travail Figma :

<https://www.figma.com/design/VI9S0LeOa9ntslPhafRVX3/Maquettes-Hotel-WPF-WEB?m=dev&node-id=0-1&t=SXObBTpinb2IDSXV-1>

3.4.1.1 Maquettes application

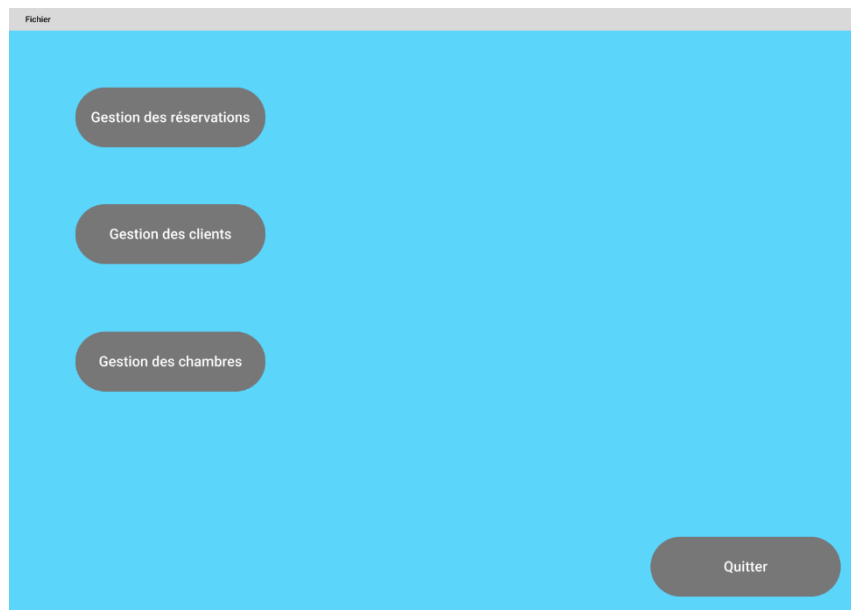


Figure 9, menu principal

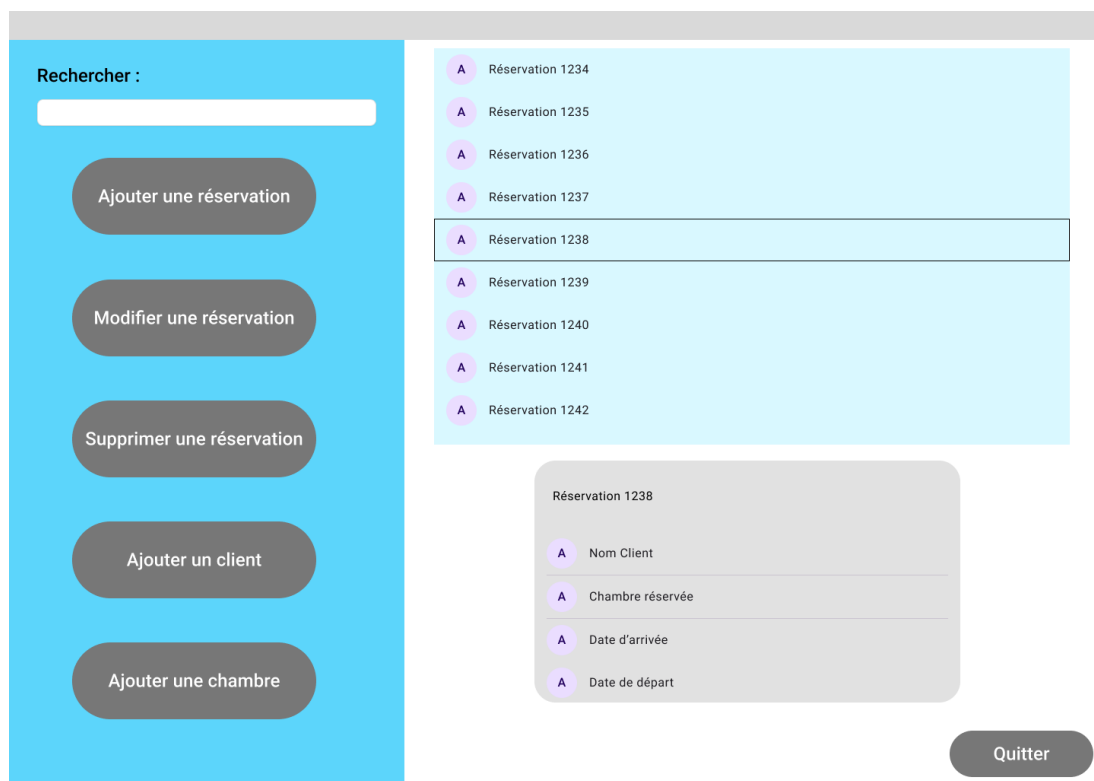


Figure 10, list view des réservations

Réserve

Client :

▼

Chambre :

▼

Nombre de personne :

Date d'arrivée :

Date de départ :

Petit-déjeuner inclus :

☐

Enregistrer

Annuler

Figure 12, formulaire d'insertion des réservations

Clients

Nom :

Prénom :

Rue :

Localité :

▼

Date de naissance :

Téléphone :

Email :

Enregistrer

Annuler

Figure 11, formulaire d'insertion client

3.4.1.2 Maquette site Web

Réservation chambre

Nom :

Prénom :

Rue :

Localité :

NPA :

Téléphone :

Nombre de personnes :

Veillez choisir une chambre :

Date d'arrivée :

Date de départ :

Petit-déjeuner inclus :

Réserver

Quitter

Figure 13, formulaire de réservation web

4 Réalisation

Pour la réalisation, nous avons plusieurs fonctionnalités différentes pour la base de données, la gestion globale dans l'application et le formulaire web.

4.1 MySQL WorkBench

Nous avons utilisé MySQL Workbench avec un serveur d'hébergement Infomaniak pour créer les tables de l'application en réalisant un ERD. Grâce à la fonctionnalité de forward engineering, nous avons généré automatiquement les tables et leurs relations.

4.1.1 ERD

Dans MySQL Workbench, nous avons créé un ERD et mis en place un trigger pour enregistrer la date et l'heure de chaque réservation de client. Les index sur les clés étrangères sont créés automatiquement et nous les avons simplement renommés. Nous n'avons pas créé d'autres index car la vitesse de recherche était suffisante. Enfin, nous avons appliqué une contrainte d'unicité sur les codes d'étages pour éviter les doublons.

Ensuite, grâce au forward engineering, les tables et les relations entre elles ont été générées automatiquement. Nous avons testé la base de données en insérant, modifiant et supprimant des données pour vérifier son bon fonctionnement. Des données ont été insérées afin de disposer de contenu pour la création de l'application.

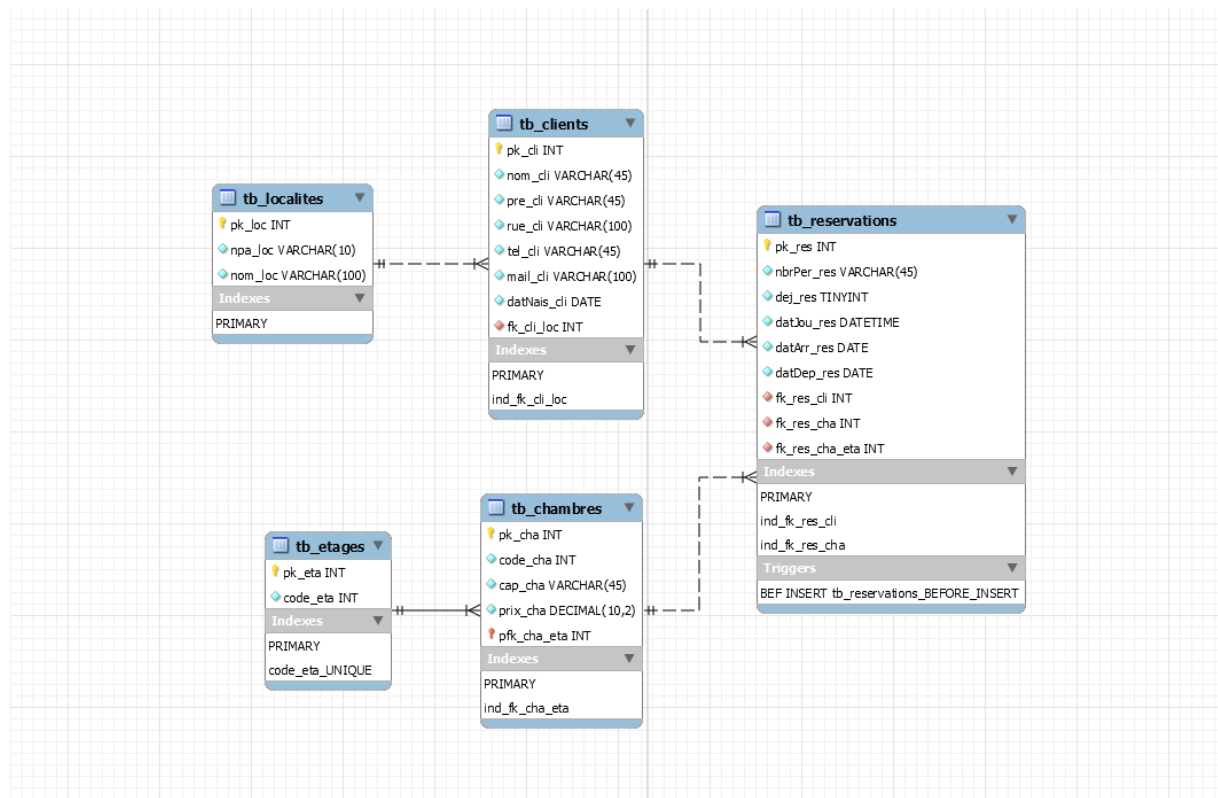


Figure 14, ERD MySQL

4.1.2 Détail du code particulier

Le code particulier notable est le trigger utilisé pour enregistrer la date et l'heure des réservations :

```

CREATE TRIGGER before_reservation_insert
BEFORE INSERT ON reservations
FOR EACH ROW
SET NEW.date_reservation = NOW();
    
```

Ce trigger s'assure que chaque nouvelle réservation a une date et une heure enregistrées automatiquement.

4.1.3 Protocoles de tests MySQL

Pour garantir le bon fonctionnement de la base de données et de l'application, plusieurs tests ont été effectués. Le protocole de test inclut l'insertion, la modification et la suppression de données pour vérifier que toutes les opérations fonctionnent correctement.

Test à effectuer		Résultat escompté	Résultat obtenu	Constatation
Localité, <i>tb_localites</i>				
1	Insertion d'une localité INSERT INTO tb_localites (npa_loc, nom_loc) VALUE('0001', 'Loc-La-LaVille');	Insère les données et génère une clé primaire	Le NPA et la localité sont insérées	Réussi
	Insertion d'une localité sans NPA INSERT INTO tb_localites (nom_loc) VALUE('Bressaucourt');	Ne devrait pas fonctionner, car il manque la localité	Fonctionne et insert un « espace » dans le champ npa_loc	Réussi
	Effacer une localité DELETE FROM tb_localites WHERE pk_loc = 110;	Efface toute la ligne ('0001', 'Loc-La-LaVille')	La ligne a disparue	Réussi
Client, <i>tb_clients</i>				
2	Insertion d'un client INSERT INTO tb_clients (nom_cli, pre_cli, rue_cli, tel_cli, mail_cli, datNais_cli, fk_cli_loc) VALUE ('Fugain', 'Michel', '9 Place de la Gare ', '+33 6 44 66 88 99', 'michel.fugain@gmail.com', '1942-05-12', 112) ;	Insère les données et génère une clé primaire	Le nom, prénom, rue, téléphone, email, date de naissance, la fk de la localité doivent être ajoutés à la table tb_clients et une clé primaire doit être généré	Réussi

	Effacer un client DELETE FROM tb_clients WHERE pk_cli = 238 ;	Efface un client	Efface toute	Réussi
	Modifier le nom d'un client UPDATE tb_clients SET nom_cli = 'Boss' WHERE pk_cli = 236;	Le nom de famille du client doit être modifié	Le nom de famille du client 236 doit être « Boss »	Réussi
Etages, tb_etages				
3	Insertion d'un étage INSERT INTO tb_etages(code_cha) VALUE (8);	Insère un étage et génère une clé primaire	L'étage a été inséré et la clé primaire générée	Réussi
	INSERT INTO tb_etages(code_cha) VALUE ('a');	Ne devrait pas fonctionner, car ce n'est pas un nombre	Le système empêche l'insertion de la lettre dans le champ	La contrainte fonctionne ➔ Réussi
	Effacer un étage DELETE FROM tb_etages WHERE code_eta = 8;	Ne doit pas fonctionner, car cela effacerait toutes les chambres	Le système empêche la suppression du 8-ème étage	La contrainte d'unicité fonctionne, il faut faire un DELETE CASCADE pour que ça marche Réussi
Chambres, tb_chambres				
4	Insertion de 2 chambres identiques INSERT INTO tb_chambres(code_cha, cap_cha, prix_cha,pfk_cha_eta)VALUES (4,3,200,8), (4,4,250,8);	Ne devrait pas fonctionner car il y aurait 2 fois la chambre 84 (pfk_cha_eta = 8 (l'étage) et code_cha = 4 (le numéro de la porte) qui for le numéro 84	Ne doit pas accepter cette insertion	Le système empêche l'insertion ➔ Réussi
	Modification du prix d'une chambre UPDATE tb_chambres SET prix_cha = 120 WHERE pk_cha = 16;	Le prix de la chambre n°11 (pk 16) doit être de CHF 120 à la place de CHF 100.-.	Le prix de la chambre n°11 est à CHF 120.-.	Réussi
Réservation, tb_reservations				

5	Insérer une réservation INSERT INTO tb_reservations(nbrPer_res, dej_res, datArr_res, datDep_res, fk_res_cli, fk_res_cha, fk_res_cha_eta) VALUE (6, 0, '2024-08-13', '2024-08-15', 237, 88, 6);	Insère les données et génère une clé primaire	La chambre numéro 66 devrait être réservé pour Michel Fugain (pk 237)	Réussi
	Insérer une réservation avec la mauvaise liaison des pk et fk. INSERT INTO tb_reservations(nbrPer_res, dej_res, datArr_res, datDep_res, fk_res_cli, fk_res_cha, fk_res_cha_eta) VALUE (6, 0, '2024-08-13', '2024-08-15', 237, 88, 1);	Ne devrait pas fonctionner, car la liaison naturelle entre la les 2 clés étrangère n'existe pas	Le système empêche l'insertion de cette donnée car la liaison entre fk_res_cha 88 et fk_res_cha_eta 1 n'existe pas	Le système empêche l'insertion ➔ Réussi
	Afficher les informations concernant une réservation SELECT pk_res AS 'Numéro de réservation', CASE WHEN code_cha < 10 THEN CONCAT(pfk_cha_eta, '-', '0', code_cha) ELSE CONCAT(pfk_cha_eta, '-', ' ', code_cha) END AS 'N°Chambre', nbrPer_res AS 'Nombre de personnes', nom_cli AS 'Nom', pre_cli AS 'Prénom', nom_loc AS 'Localité', datArr_res AS 'Arrivée', datDep_res AS 'Départ' FROM ih1o2_pi02db3.tb_reservations INNER JOIN ih1o2_pi02db3.tb_clients ON fk_res_cli = pk_cli INNER JOIN ih1o2_pi02db3.tb_chambres ON pk_cha = fk_res_cha INNER JOIN ih1o2_pi02db3.tb_localites ON pk_loc = fk_cli_loc LEFT JOIN ih1o2_pi02db3.tb_etages ON pfk_cha_eta = pk_eta;	Doit afficher les informations provenant de la table tb_reservations ainsi que des tables associées par des jointures.	Affiche les informations demandées en requête	Réussi

4.2 Gestion des clients

Explication détaillée de la fonctionnalité, du cas d'utilisation

Dans l'application WPF, nous avons une fonctionnalité de gestion des clients qui consiste à pouvoir modifier, supprimer et ajouter des clients. En premier lieu, il y a une liste view qui affiche tous les clients dans la base de données, ce qui permet d'avoir une vue globale de tous les clients. En cliquant sur un client, un sous-formulaire s'affiche avec toutes les réservations qu'un client a effectué. Puis, nous avons ajouté des boutons qui nous permettent de faire des actions de manipulation de données.

Le bouton « Ajouter » ouvre un formulaire vide pour ajouter les informations nécessaires à la création du client dans la base de données. Le bouton « Modifier » ouvre le même formulaire mais reprend les informations du client sélectionné pour permettre la modification dans la base. Le bouton « Supprimer » affiche une demande de confirmation de suppression avant de supprimer définitivement le client de la base de données.

Les formulaires ont également trois boutons « Enregistrer », « Retour » et « Ajouter une localité » pour enregistrer les données, fermer le formulaire sans enregistrement ou ajouter une localité si celle-ci, ne se trouve déjà pas dans la liste déroulante du formulaire.

Figure 15, formulaire d'insertion client

4.2.1 Nouveau client

Déroulement ou explication d'algorithmes particuliers

L'employé de l'hôtel va dans la gestion des clients depuis l'écran d'accueil. Il clique sur le bouton « Ajouter » pour ouvrir le formulaire d'ajout client puis remplit les champs. Tous les champs sont obligatoires et il y a un contrôle de saisi pour la date de naissance, le client doit être majeur pour réserver une chambre d'hôtel. L'employé doit sélectionner une localité dans la liste déroulante et peut en ajouter une si celle-ci ne s'y trouve pas. Une fois les champs remplis, il clique sur « Enregistrer » ce qui va insérer le client dans la base de données.

4.2.2 Détail du code particulier

Grâce au ClientRepository, les modifications, suppressions et ajouts des clients sont enregistrés dans la base de données.

4.2.3 Protocoles de tests

Test à effectuer		Résultat Escompté	Résultat Obtenu	Constatation
Insérer un client				
1	Ajouter un nouveau client	Un nouveau client doit 'etre inséré dans la base de donnée	Le client est dans la base de donnée	Réussi

	Ajouter un client qui a moins de 18 ans	Ne devrait pas fonctionner, car un client ne peut pas avoir moins de 18 ans	Message box demande d'insérer un age entre 18 et 105 ans	Réussi
	Ajouter un client avec un champs vide	Ne devrait pas fonctionner	N'autorise pas un champ nul	Réussi
Supprimer				
2	Supprimer un client qui n'a pas de réservation	Devrait supprimer le client de la base de données	Supprime le client de la base de données	Réussi
	Supprimer un client qui a une réservation	Ne doit pas fonctionner	N'autorise pas la suppression d'un client qui a une réservation	Réussi
Modification d'un client				
3	Modifier les informations de tous les champs d'un client	Devrait changer toutes les informations dans la base de donnée	Change toutes les informations dans la base de donnée	Réussi

4.3 Gestion des réservations

La fonctionnalité de gestion des réservations permet la manipulation des données des réservations de chambres d'hôtel. En premier, il y a une liste view qui permet de voir toutes les réservations avec leur numéro, la chambre réservée, la date d'arrivée et la date de départ. En cliquant sur une réservation, le détail de la réservation s'affiche avec le nom et prénom du client et ses coordonnées.

Pour ajouter une réservation, il y a un bouton « Ajouter une réservation » qui ouvre un formulaire vide à remplir pour créer une réservation de chambre, un bouton « Modifier une réservation » qui en sélectionnant une réservation va reprendre les informations dans le formulaire et permettre la mise à jour des données. Puis, un bouton « Supprimer une réservation » qui supprime de la base de données la réservation.

4.3.1 Nouvelle réservation

L'employé de l'hôtel va dans la gestion des réservations depuis l'écran d'accueil. Il clique sur « Ajouter une réservation » et choisit dans la liste déroulante le client concerné. Puis, choisit la chambre à réserver dans une liste déroulante également. Ensuite, il insère le nombre de personne dans la réservation et la date d'arrivée et de départ et si les clients veulent un petit-déjeuner inclus

dans leur réservation. Une fois le formulaire rempli, le bouton « Enregistrer » va enregistrer les données dans la base et l'afficher dans la list view.



Figure 16, list view des réservations

4.3.2 Détail du code particulier

Grâce au ReservationRepository, les modifications, suppressions et ajouts sont enregistrés directement dans la base de données.

4.3.3 Protocoles de tests

Test à effectuer		Résultat Escompté	Résultat Obtenu	Constatation
Insérer une réservation				
1	Ajouter une réservation	Ajouter une nouvelle réservation dans la base de données	La réservation est ajoutée dans la base de données	Réussi
	Ajouter une réservation pour une chambre déjà occupée	Ne devrait pas fonctionner, car une chambre ne peut pas être réservée si elle est déjà occupée	Message indiquant que la chambre est déjà occupée	Réussi
	Ajouter une réservation avec un champ vide	Ne devrait pas fonctionner	N'autorise pas un champ nul	Réussi
Supprimer une réservation				
2	Supprimer une réservation	Supprimer une réservation de la base de données	Supprime le client de la base de données	Réussi

	Supprimer un client qui a une réservation	Ne doit pas fonctionner	N'autorise pas la suppression d'un client qui a une réservation	Réussi
Modification une réservation				
3	Modifier une réservation	Modifier tous les champs d'une réservation	Change toutes les informations dans la base de données	Réussi

4.4 Gestion des chambres

La fonctionnalité de gestion des chambres va permettre de gérer les chambres de l'hôtel. Il y a une list view qui s'affiche avec toutes les chambres et des boutons sur le côté qui permettent de supprimer, ajouter ou modifier une chambre. Pour ajouter une chambre, on clique sur le bouton « Ajouter » ce qui va ouvrir un formulaire vide pour ajouter une chambre. Il y a une liste déroulante des étages, le prix et la capacité (nombre de personne pouvant être accueilli dans la chambre) et le numéro de la chambre. Le bouton « Modifier » permet la modification d'une chambre en reprenant les informations dans le formulaire. Le bouton « Supprimer » efface la chambre dans la base de données.

4.4.1 Ajout d'une chambre

L'employé de l'hôtel peut accéder à la gestion des chambres depuis l'écran d'accueil. Pour ajouter une chambre, il saisit le prix et la capacité, puis sélectionne l'étage correspondant dans une liste déroulante. Le numéro de chambre, étant attribué automatiquement, se suit naturellement après les autres numéros de chambre. Le numéro de chambre est formé par le numéro d'étage suivi du numéro de la chambre. Par exemple, la 11ème chambre au 2ème étage sera numérotée 2-11. Après enregistrement du formulaire, la chambre créée est visible dans la liste de gestion des chambres.

Pour modifier le prix d'une chambre, l'employé sélectionne la chambre concernée, puis clique sur « Modifier une chambre », ouvrant ainsi un formulaire pré-rempli avec les informations actuelles de la chambre. Il peut alors ajuster le prix et cliquer sur « Enregistrer » pour mettre à jour les données.

4.4.2 Détail du code particulier

Grâce au ChambreRepository, les modifications, suppressions et ajouts sont enregistrés dans la base de données.

4.4.3 Protocoles de tests

Test à effectuer		Résultat Escompté	Résultat Obtenu	Constatation
Insérer une chambre				
1	Ajouter une chambre	Ajouter une nouvelle chambre dans la base de données et générer un numéro de chambre avec le numéro de l'étage devant.	La chambre est ajoutée dans la base de données	Réussi
Supprimer d'une chambre				
2	Supprimer une chambre	Supprimer une chambre de la base de données	Supprime la chambre de la base de données	Réussi
Modification d'une chambre				
3	Modifier une chambre	Modifier tous les champs d'une réservation	Change toutes les informations dans la base de données, mais parfois l'application crash après cette opération	Réussi, mais il faudrait trouver quelle est la cause de ces crashes

4.5 Site Web

Le site web de l'hôtel est une fonctionnalité qui va permettre aux clients de contacter l'hôtel grâce au formulaire de contact. Celui-ci comprend une page d'accueil avec un carrousel de photo qui défilent et un petit texte de bienvenue. Ensuite, il y a une page de contact où se trouve le formulaire de contact à remplir par le client pour être recontacté un employé de l'hôtel.

Le site est conçu pour être « responsive », s'adaptant automatiquement à différents types d'écrans.

Lien : <https://lmo.divtec.me/2023/pi02-grpe3>

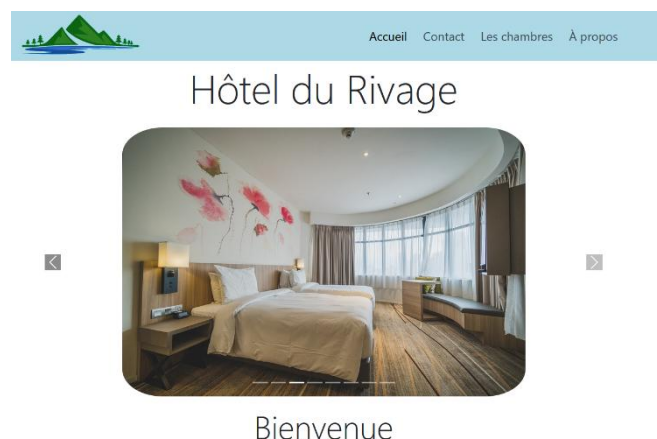


Figure 17, page d'accueil du site web

4.5.1 Fonctionnalités Principales

Le site web comprend les fonctionnalités suivantes :

- **Formulaire de Contact** : Les visiteurs peuvent remplir un formulaire de contact pour entrer en communication avec notre équipe. Ce formulaire permet de collecter les informations essentielles telles que le nom, l'adresse e-mail et le message du client, assurant ainsi une interaction fluide et directe.
- **Présentation de l'Hôtel** : Le site offre une interface conviviale permettant aux utilisateurs de découvrir les caractéristiques de notre hôtel. Cela inclut des sections dédiées aux chambres disponibles, aux installations comme les restaurants, salles de réunion, piscines ou spas, et aux activités locales.

4.5.2 Fonctionnalités Clés

Le site web inclut les fonctionnalités suivantes, avec un accent particulier sur le contrôle des saisies utilisateur :

- **Formulaire de Contact Sécurisé** : Nous avons mis en place un formulaire de contact sécurisé qui guide les utilisateurs à travers la saisie de leurs informations personnelles, telles que le nom, l'adresse e-mail et le message. Chaque champ du formulaire est validé côté client et côté serveur pour garantir que les données soumises sont correctes et complètes.
- **Validation Côté Client** : Utilisation de JavaScript pour effectuer une validation instantanée des champs du formulaire avant soumission. Cela inclut la vérification de la syntaxe de l'adresse e-mail, la longueur minimale des messages, et d'autres critères pertinents.
- **Validation Côté Serveur** : Une fois que les données sont soumises, elles sont validées à nouveau côté serveur pour prévenir toute tentative de soumission de données incorrectes ou malveillantes. Cette étape de validation renforce la sécurité du formulaire et garantit l'intégrité des données collectées.

4.5.3 Technologies Utilisées

Pour assurer une expérience utilisateur optimale, nous avons choisi les technologies suivantes :

- **Frontend** : HTML5, CSS3 avec de le Framework Bootstrap, JavaScript
- **Backend** : PHP (pour le traitement des formulaires et la gestion des données côté serveur)
- **Base de données** : MySQL (pour le stockage sécurisé des données des clients et des informations sur l'hôtel)

4.5.4 Formulaire de contact

Le client remplit un formulaire de contact, et les informations sont directement intégrées dans la base de données. Si certains champs ne sont pas remplis, une alert apparaît indiquant quel/quels champs l'utilisateur doit remplir.



Formulaire de contact

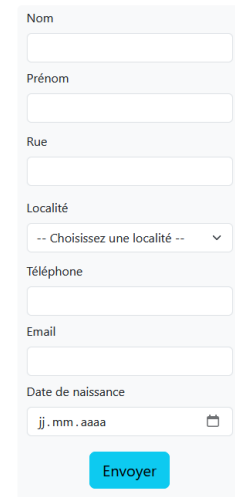


Figure 18, formulaire de contact web

4.5.5 Détail du code particulier

Le client remplit un formulaire de contact, et les informations sont directement intégrées dans la base de données. En cas de champs non remplis, une alerte apparaît pour indiquer à l'utilisateur les champs qu'il doit compléter.

4.5.6 Protocoles de tests

Insertion des données : Vérifiez dans MySQL Workbench et dans l'application si les informations sont correctement enregistrées. Tout est bien enregistré, confirmant que le système de base de données est centralisé. L'utilisateur est ensuite redirigé vers la page d'accueil.

Insertion de données avec plusieurs champs vides : Une alerte s'affiche pour informer l'utilisateur qu'il ne peut pas laisser de champs vides. L'envoi du formulaire est annulé.

Figure 19, formulaire de contact

4.5.7 QR code

Nous avons créé un QR code qui peut être scanné avec un téléphone portable pour accéder directement au site web de l'Hôtel du Rivage. Il envoie l'utilisateur sur <https://lmo.divtec.me/2023/pi02-grpe3>.



Figure 20, QR code lien site internet



Figure 21, page d'accueil téléphone portable

5 Livraison

5.1 Mode d'emploi utilisateur

Les modes d'emploi sont en annexes dans le dossier Doc.

6 Conclusions

6.1 Conclusion Maëlle

Ce projet m'a permis de mettre en pratique plusieurs cours que l'on a vu cette année notamment la modélisation de base de données, le développement C# et le développement Web. Je suis contente du projet qu'on a mené et le résultat correspond à mes attentes.

J'ai apprécié coder en C# car cela m'a permis de bien comprendre comment développer une application en utilisant l'architecture MVVM. On a eu quelques challenges durant le développement avec les clés naturelles des chambres et des étages mais avec de l'aide, on a réussi à résoudre nos problèmes.

Le site web était très intéressant à faire également car nous avons utilisé le framework Bootstrap. C'est un framework que je n'avais jamais utilisé donc j'ai dû apprendre à l'utiliser. Au final, il est plutôt simple à utiliser une fois qu'on a compris et ça nous a permis de faire un site web complet avec un visuelle agréable et le responsive ce fait automatiquement.

Ce projet m'a beaucoup appris de ce qui nous attend dans notre futur métier d'informaticien et c'était très intéressant créer une application complète de A à Z et un site web à côté.

6.2 Conclusion Alexandre

J'ai apprécié développer mes compétences acquises au cours de cette année. Codé en ViewModel en étant connecté à une base de données était très intéressant. J'ai pris beaucoup de plaisir à utiliser MySQL Workbench, où j'ai pu mettre en pratique des concepts vus en cours, tels que les bases de données, les schémas de base de données, et l'ajout d'OCL. Je suis également content d'avoir acquis une nouvelle compétence en apprenant à utiliser Bootstrap, un framework facile à prendre en main lorsque l'on est à l'aise en CSS. Même si cela reste un apprentissage, c'est vraiment pratique pour le responsive design.

J'ai aimé travailler sur cette application WPF en C#. Il y a eu beaucoup de défis, notamment avec les entités naturelles dépendantes et les PFK. Nous voulions ajouter plus d'options, mais j'ai réalisé que les problèmes de code peuvent parfois stopper le projet pendant longtemps. Finalement, nous avons dû nous concentrer sur ce que nous pouvions développer dans le temps imparti. Je suppose que pour des projets plus grands, comme des ERP, cela nécessite une organisation rigoureuse, une bonne conception, et une répartition efficace des tâches. Cela doit être complexe, nécessitant assiduité et rigueur.

Quoi qu'il en soit, j'ai pris beaucoup de plaisir durant cet atelier, et cela m'a donné un avant-goût du monde du travail en informatique. J'avais presque l'impression d'être en situation professionnelle, c'était vraiment enrichissant.

7 Annexes

7.1 Code source

Le code du projet est disponible sur GitHub : [2023-2025-c822-groupe3](https://github.com/2023-2025-c822-groupe3).

Dans le dossier **Web**, vous trouverez les fichiers relatifs à la page web, et dans le dossier **Windows**, ceux relatifs à l'application.

7.2 Source

- Canal 122 et 822 des équipes Teams
- <https://developpez.net/>
- <https://www.microsoft.com/>
- <https://www.youtube.com/>
- <https://chatgpt.com/>
- <https://getbootstrap.com/>
- <https://dev.mysql.com/doc/>
- <https://bootstrap21.org/fr/>
- <https://www.w3schools.com/>
- <https://sql.sh/>

7.3 Table des illustrations

Figure 1, arborescence dossier Microsoft Teams	6
Figure 2, Gantt prévisionnel partie 1.....	7
Figure 3, Gantt prévisionnel partie 2.....	8
Figure 4, Gantt final partie 1	9
Figure 5, Gantt final partie 2	10
Figure 6, architecture du système de réservation	11
Figure 7, modèle conceptuel de données	13
Figure 8, modèle logique de données	15
Figure 9, menu principal.....	16
Figure 10, list view des réservations	16
Figure 11, formulaire d'insertion client.....	17
Figure 12, formulaire d'insertion des réservations	17
Figure 13, formulaire de réservation web.....	18
Figure 14, ERD MySQL	19
Figure 15, formulaire d'insertion client.....	23
Figure 16, list view des réservations	25
Figure 17, page d'accueil du site web	27
Figure 18, formulaire de contact web	28
Figure 19, formulaire de contact	29
Figure 20, QR code lien site internet.....	29
Figure 21, page d'accueil téléphone portable.....	29