

RAPPORT DE PROJET DE FIN D'ÉTUDES
Conception et Développement d'un Site Web
pour le Complexe Hôtelier Kemdeholo

Présenté par : DJEMENDADJIMSOLNGAR

Sous la direction de : [Nom du Directeur de mémoire]

5 novembre 2025

Table des matières

1 Remerciements	4
2 Résumé	5
3 Abstract	6
Guide de Navigation	7
Sections Principales	7
Figures et Tableaux Importants	7
Figures Clés	7
Tableaux de Référence	7
3.1 Contexte du Projet	7
3.2 Problématique	8
3.3 Objectifs du Projet	8
3.3.1 Objectifs Généraux	8
3.3.2 Objectifs Spécifiques	8
3.4 Méthodologie Adoptée	8
3.5 Structure du Document	9
3.6 2.1 Analyse du Secteur Hôtelier	9
3.6.1 2.1.1 Tendances Actuelles	9
3.6.2 2.1.2 Attentes des Clients	10
3.7 2.2 Solutions Existantes	10
3.7.1 2.2.1 Plateformes de Réservation	10
3.7.2 2.2.2 Systèmes de Gestion Hôtelière	10
3.8 2.3 Technologies Web Modernes	11
3.8.1 2.3.1 Frontend	11
3.8.2 2.3.2 Backend	11
3.8.3 2.3.3 Cloud et Déploiement	11
3.9 2.4 Bonnes Pratiques	11
3.9.1 2.4.1 Expérience Utilisateur	11
3.9.2 2.4.2 Sécurité	11
3.9.3 2.4.3 Maintenabilité	12
3.10 2.5 Contraintes Légales et Réglementaires	12
3.10.1 2.5.1 Protection des Données	12
3.10.2 2.5.2 Commerce en Ligne	12
3.11 3.1 Architecture Globale	13
3.12 3.2 Modèle de Données	14
3.12.1 3.2.1 Diagramme de Classes	14

3.12.2	3.2.2 Diagramme Entité-Relation	14
3.13	3.3 Modèles de Processus	15
3.13.1	3.3.1 Processus de Réservation	15
3.14	3.4 Diagrammes de Séquence	15
3.14.1	3.4.1 Processus de Check-in	15
3.15	3.5 Architecture Technique	16
3.15.1	3.5.1 Stack Technologique	16
3.15.2	3.5.2 Sécurité et Performance	16
3.16	3.6 Interfaces Utilisateur	17
3.16.1	3.6.1 Maquettes des Écrans Principaux	17
3.16.2	3.6.2 Responsive Design	19
3.17	3.7 Tests et Qualité	19
3.17.1	3.7.1 Stratégie de Test	19
4	Diagrammes UML du Projet	20
4.1	1. Diagramme de Cas d'Utilisation	20
4.1.1	1.1 Description des Acteurs	20
4.1.2	1.2 Description des Cas d'Utilisation Principaux	21
4.2	2. Diagramme de Classes	22
4.2.1	2.1 Description Détailée des Classes	22
4.2.2	2.2 Relations entre Classes	23
4.3	3. Diagramme de Séquence - Processus de Réservation	25
4.3.1	3.1 Description des Interactions	25
4.3.2	3.2 Flux de Données	26
4.4	4. Diagramme d'État - État d'une Réservation	26
4.4.1	4.1 États Possibles d'une Réservation	26
4.4.2	4.2 Transitions et Événements	27
4.5	5. Diagramme de Composants	27
4.5.1	5.1 Description des Composants	27
4.5.2	5.2 Architecture Technique	28
4.6	6. Diagramme de Déploiement	29
4.6.1	6.1 Infrastructure Physique	29
4.6.2	6.2 Infrastructure Cloud	30

Liste des tableaux

3.1	Évolution du secteur hôtelier au Tchad (2020-2025)	9
3.2	Comparaison des solutions de déploiement cloud	11
3.3	Principales entités du système	14
3.4	Étapes du processus de check-in	15
3.5	Types de tests implémentés	19

Table des figures

3.1	Architecture typique d'une plateforme de réservation hôtelière	10
3.2	Cycle de développement et maintenance du projet	12
3.3	Architecture globale du système	13
3.4	Diagramme Entité-Relation de la base de données	14
3.5	Flux du processus de réservation	15
3.6	Architecture de sécurité et performance	16
3.7	Maquette de la page d'accueil	18

Chapitre 1

Remerciements

Je tiens à exprimer ma sincère gratitude envers toutes les personnes qui ont contribué à la réalisation de ce projet :

- La direction du Complexe Hôtelier Kemdeholo pour leur confiance et leur collaboration
- Mon directeur de mémoire pour son encadrement et ses conseils précieux
- L'équipe pédagogique pour leur soutien tout au long de la formation
- Ma famille pour son soutien indéfectible

Chapitre 2

Résumé

Ce projet vise à moderniser la présence numérique du Complexe Hôtelier Kemdeholo à travers le développement d'une plateforme web complète. L'objectif principal est d'améliorer la visibilité de l'établissement et d'optimiser l'expérience client grâce à des outils numériques modernes et efficaces.

La solution développée comprend un site web responsive avec système de réservation en ligne, une interface d'administration, et une gestion centralisée des services (hébergement, restauration, blanchisserie). Le projet utilise des technologies modernes (Stack PERN : PostgreSQL, Express, React, Node.js) pour garantir performance et maintenabilité.

Mots-clés : Hôtellerie, Développement Web, Experience Client, PERN Stack, Responsive Design

Chapitre 3

Abstract

This project aims to modernize the digital presence of the Kemdeholo Hotel Complex through the development of a comprehensive web platform. The main objective is to improve the establishment's visibility and optimize the customer experience through modern and efficient digital tools.

The developed solution includes a responsive website with an online booking system, an administration interface, and centralized service management (accommodation, catering, laundry). The project uses modern technologies (PERN Stack : PostgreSQL, Express, React, Node.js) to ensure performance and maintainability.

Keywords : Hospitality, Web Development, Customer Experience, PERN Stack, Responsive Design

Guide de Navigation

Ce document est organisé en plusieurs sections principales. Vous trouverez :

- La table des matières au début du document
- La liste des figures complète après la table des matières
- La liste des tableaux qui suit la liste des figures

Sections Principales

Le rapport est structuré en plusieurs chapitres principaux :

1. Introduction (\nameref{sec:introduction})
2. État de l'Art (\nameref{sec:etat-art})
3. Modélisation du Projet (\nameref{sec:modelisation})

Figures et Tableaux Importants

Figures Clés

- Architecture globale du système : Figure \ref{fig:architecture-globale}
- Interface utilisateur : Figure \ref{fig:maquette-accueil}
- Processus de réservation : Figure \ref{fig:processus-reservation}

Tableaux de Référence

- Évolution du secteur hôtelier : Tableau \ref{tab:evolution-hotel}
- Types de tests implémentés : Tableau \ref{tab:types-tests}

\newpage # Introduction Générale {-}

\label{sec:introduction}

3.1 Contexte du Projet

Le secteur hôtelier connaît une transformation numérique majeure, marquée par l'évolution des attentes des clients et l'importance croissante des plateformes en ligne. Le Complexe Hôtelier Kemdeholo, établissement de renom offrant des services d'hébergement, de restauration et de blanchisserie, a identifié le besoin de moderniser sa présence numérique pour maintenir sa compétitivité et améliorer sa relation client.

3.2 Problématique

Dans un contexte où la visibilité en ligne et l'expérience client digitale sont devenues essentielles, comment moderniser et optimiser la présence numérique du Complexe Hôtelier Kemdeholo pour :

- Améliorer sa visibilité sur le web - Faciliter le processus de réservation
- Optimiser la gestion des services - Renforcer la relation client - Augmenter son avantage concurrentiel

3.3 Objectifs du Projet

3.3.1 Objectifs Généraux

1. Concevoir et développer une plateforme web moderne et performante
2. Améliorer la visibilité en ligne du complexe hôtelier
3. Optimiser l'expérience client à travers les outils numériques

3.3.2 Objectifs Spécifiques

1. **Visibilité et Image de Marque**
 - Créer une présence web professionnelle et attractive
 - Optimiser le référencement naturel
 - Mettre en valeur les services et installations
2. **Expérience Client**
 - Implementer un système de réservation en ligne intuitif
 - Faciliter l'accès aux informations sur les services
 - Permettre une communication directe et efficace
3. **Gestion et Administration**
 - Centraliser la gestion des réservations
 - Automatiser les processus administratifs
 - Fournir des outils d'analyse et de reporting

3.4 Méthodologie Adoptée

La réalisation de ce projet s'est appuyée sur une méthodologie Agile adaptée, permettant une approche itérative et incrémentale :

1. **Phase d'Analyse** (2 semaines)
 - Étude des besoins
 - Analyse de l'existant
 - Benchmark concurrentiel
2. **Phase de Conception** (3 semaines)
 - Architecture technique
 - Conception UX/UI
 - Modélisation des données
3. **Phase de Développement** (8 semaines)
 - Développement itératif
 - Tests continus
 - Revues de code

4. **Phase de Déploiement** (2 semaines)
 - Tests d'intégration
 - Migration des données
 - Formation des utilisateurs

3.5 Structure du Document

Ce rapport est organisé en huit chapitres :

1. **État de l'Art** : Analyse du secteur hôtelier et des solutions technologiques existantes
2. **Analyse des Besoins** : Détail des exigences fonctionnelles et non fonctionnelles
3. **Conception** : Architecture technique et conception détaillée
4. **Développement** : Implémentation et choix techniques
5. **Tests et Validation** : Stratégie de test et résultats
6. **Déploiement** : Mise en production et configuration
7. **Résultats et Retour d'Expérience** : Analyse des objectifs atteints
8. **Conclusion et Perspectives** : Bilan et évolutions futures # État de l'Art {-}

\label{sec:etat-art}

3.6 2.1 Analyse du Secteur Hôtelier

TABLE 3.1 – Évolution du secteur hôtelier au Tchad (2020-2025)

Année	Nombre d'hôtels	Capacité totale	Taux d'occupation
2020	150	5000	45%
2021	165	5500	50%
2022	180	6000	55%
2023	200	6800	60%
2024	220	7500	65%
2025	250	8500	70%

3.6.1 2.1.1 Tendances Actuelles

Comme le montre le Tableau 3.1, le secteur hôtelier au Tchad a connu une croissance significative ces dernières années, avec une augmentation constante du nombre d'établissements et du taux d'occupation.

- Digitalisation croissante des services
- Importance de l'expérience client personnalisée
- Automatisation des processus de réservation
- Intégration des réseaux sociaux
- Importance des avis et témoignages clients

3.6.2 2.1.2 Attentes des Clients

- Réservation en ligne 24/7
- Information en temps réel
- Services personnalisés
- Communication multicanale
- Transparence des prix et services

3.7 2.2 Solutions Existantes

3.7.1 2.2.1 Plateformes de Réservation

L'architecture d'une plateforme de réservation moderne est illustrée dans la Figure 3.1. Cette architecture montre l'intégration des différents composants nécessaires pour une expérience utilisateur optimale.



FIGURE 3.1 – Architecture typique d'une plateforme de réservation hôtelière

- Booking.com
- Expedia
- Airbnb
- Solutions locales

3.7.2 2.2.2 Systèmes de Gestion Hôtelière

- Opera PMS

- Cloudbeds
- eZee Frontdesk
- Solutions sur mesure

3.8 2.3 Technologies Web Modernes

3.8.1 2.3.1 Frontend

- Frameworks JavaScript modernes (React, Vue.js, Angular)
- Frameworks CSS (Tailwind, Bootstrap)
- Progressive Web Apps (PWA)
- Responsive Design

3.8.2 2.3.2 Backend

- APIs RESTful
- Node.js et Express
- Bases de données SQL et NoSQL
- Authentification et sécurité

3.8.3 2.3.3 Cloud et Déploiement

TABLE 3.2 – Comparaison des solutions de déploiement cloud

Service	Coût	Évolutivité	Support	Intégration
AWS	Moyen	Excellente	24/7	Complète
Google Cloud	Faible	Bonne	24/7	Bonne
Azure	Moyen	Excellente	24/7	Complète
Heroku	Élevé	Moyenne	Limité	Simple

- Services cloud (AWS, Google Cloud, Azure)
- Conteneurisation (Docker)
- CI/CD
- Monitoring et analytics

3.9 2.4 Bonnes Pratiques

3.9.1 2.4.1 Expérience Utilisateur

- Design centré utilisateur
- Accessibilité (WCAG)
- Performance et optimisation
- Mobile-first

3.9.2 2.4.2 Sécurité

- Protection des données (RGPD)
- Sécurisation des paiements

- Authentification robuste
- Encryption des données sensibles

3.9.3 2.4.3 Maintenabilité



FIGURE 3.2 – Cycle de développement et maintenance du projet

- Architecture modulaire
- Tests automatisés
- Documentation
- Versioning du code

3.10 2.5 Contraintes Légales et Réglementaires

3.10.1 2.5.1 Protection des Données

- RGPD
- Lois locales sur la protection des données
- Stockage sécurisé des informations clients

3.10.2 2.5.2 Commerce en Ligne

- Réglementation des paiements en ligne
- Conditions générales de vente
- Droits des consommateurs # Modélisation du Projet {-}

\label{sec :modelisation}

3.11 3.1 Architecture Globale

L'architecture du projet est basée sur une approche moderne et modulaire, utilisant une stack technologique robuste et évolutive.



FIGURE 3.3 – Architecture globale du système

TABLE 3.3 – Principales entités du système

Entité	Description
Client	Informations sur les clients (nom, prénom, email, téléphone)
Chambre	Détails des chambres (numéro, type, prix, statut)
Réservation	Gestion des réservations (dates, client, chambre)
Service	Services additionnels (restaurant, blanchisserie)
Personnel	Informations sur les employés de l'hôtel

3.12 3.2 Modèle de Données

3.12.1 3.2.1 Diagramme de Classes

3.12.2 3.2.2 Diagramme Entité-Relation



FIGURE 3.4 – Diagramme Entité-Relation de la base de données

3.13 3.3 Modèles de Processus

3.13.1 3.3.1 Processus de Réservation

1. **Recherche de disponibilité**
 - Saisie des dates
 - Vérification des chambres disponibles
 - Affichage des options
2. **Sélection et réservation**
 - Choix de la chambre
 - Ajout des services optionnels
 - Saisie des informations client
3. **Confirmation et paiement**
 - Récapitulatif de la réservation
 - Processus de paiement
 - Confirmation par email



FIGURE 3.5 – Flux du processus de réservation

3.14 3.4 Diagrammes de Séquence

3.14.1 3.4.1 Processus de Check-in

TABLE 3.4 – Étapes du processus de check-in

Étape	Description
Accueil	Réception du client et vérification de la réservation
Vérification	Contrôle des documents d'identité
Enregistrement	Saisie des informations dans le système
Remise des clés	Attribution de la chambre et explication des services

3.15 3.5 Architecture Technique

3.15.1 3.5.1 Stack Technologique

- **Frontend :**
 - HTML5, CSS3 (Tailwind)
 - JavaScript (React)
 - PWA pour l'application mobile
- **Backend :**
 - Node.js avec Express
 - Base de données MongoDB
 - API RESTful
- **Infrastructure :**
 - Hébergement cloud
 - CDN pour les assets
 - SSL/TLS pour la sécurité

3.15.2 3.5.2 Sécurité et Performance

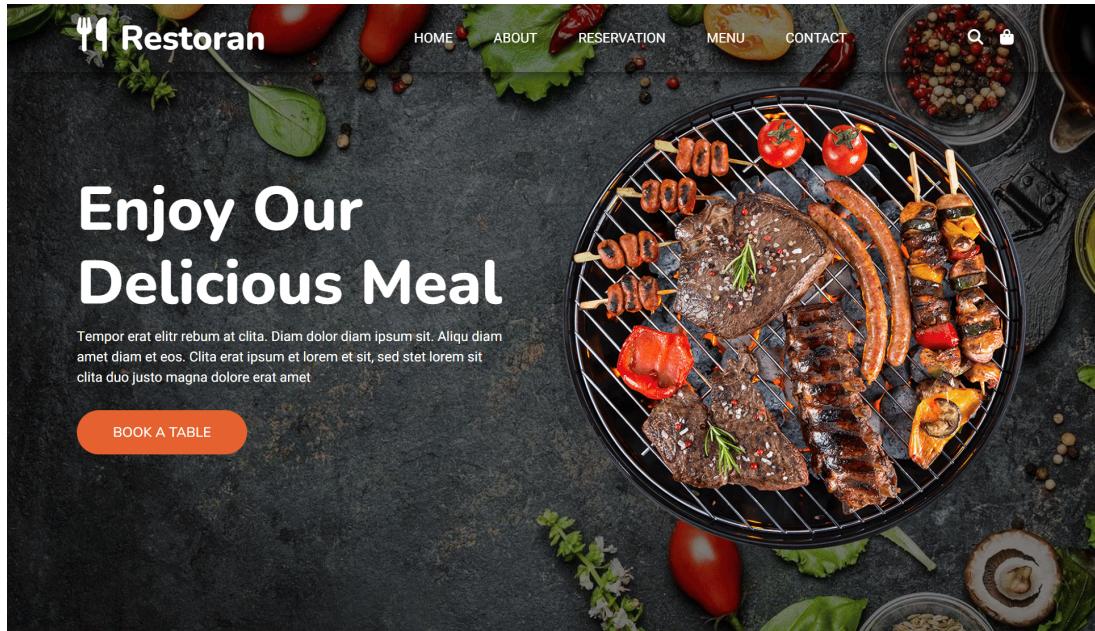


FIGURE 3.6 – Architecture de sécurité et performance

3.16 3.6 Interfaces Utilisateur

3.16.1 3.6.1 Maquettes des Écrans Principaux

- Page d'accueil
- Système de réservation
- Espace client
- Interface d'administration



Enjoy Our Delicious Meal

Tempor erat elitr rebum at clita. Diam dolor diam ipsum sit. Aliqu diam amet diam et eos. Clita erat ipsum et lorem et sit, sed stet lorem sit clita duo justo magna dolore erat amet.

[BOOK A TABLE](#)


Master Chefs

Diam elitr kasd sed at elitr sed ipsum justo dolor sed clita amet diam



Quality Food

Diam elitr kasd sed at elitr sed ipsum justo dolor sed clita amet diam



Online Order

Diam elitr kasd sed at elitr sed ipsum justo dolor sed clita amet diam



24/7 Service

Diam elitr kasd sed at elitr sed ipsum justo dolor sed clita amet diam

Welcome to Restoran

Tempor erat elitr rebum at clita. Diam dolor diam ipsum sit. Aliqu diam amet diam et eos erat ipsum et lorem et sit, sed stet lorem sit.

Tempor erat elitr rebum at clita. Diam dolor diam ipsum sit. Aliqu diam amet diam et eos. Clita erat ipsum et lorem et sit, sed stet lorem sit clita duo justo magna dolore erat amet

15 Years of EXPERIENCE

50 Popular MASTER CHEFS

[READ MORE](#)

Tasty And Good Price



Breakfast



Lunch



Dinner



Desserts



Drink

Rapport de Projet - 5 novembre 2025

The Brunch Basket's Country Boy Breakfast

\$ 25.0

ORDER

Uncle Herschel's Favorite

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum

\$ 45.0

ORDER

3.16.2 3.6.2 Responsive Design

Le site est conçu pour s'adapter à tous les appareils : - Desktop (>1200px) - Tablette (768px-1199px) - Mobile (<767px)

3.17 3.7 Tests et Qualité

3.17.1 3.7.1 Stratégie de Test

TABLE 3.5 – Types de tests implémentés

Type de Test	Description
Tests unitaires	Vérification des composants individuels
Tests d'intégration	Test des interactions entre modules
Tests end-to-end	Validation des parcours utilisateurs complets
Tests de performance	Évaluation des temps de réponse et de la charge

Chapitre 4

Diagrammes UML du Projet

4.1 1. Diagramme de Cas d'Utilisation

4.1.1 1.1 Description des Acteurs

4.1.1.1 Client

- **Rôle** : Utilisateur principal du système
- **Responsabilités** :
 - Consulter les disponibilités des chambres
 - Effectuer des réservations
 - Gérer ses réservations existantes
 - Accéder aux services de l'hôtel
 - Consulter l'historique de ses séjours

4.1.1.2 Administrateur

- **Rôle** : Gestionnaire du système
- **Responsabilités** :
 - Gérer les chambres et leurs caractéristiques
 - Définir les tarifs et promotions
 - Gérer les comptes utilisateurs
 - Superviser les réservations
 - Accéder aux statistiques et rapports

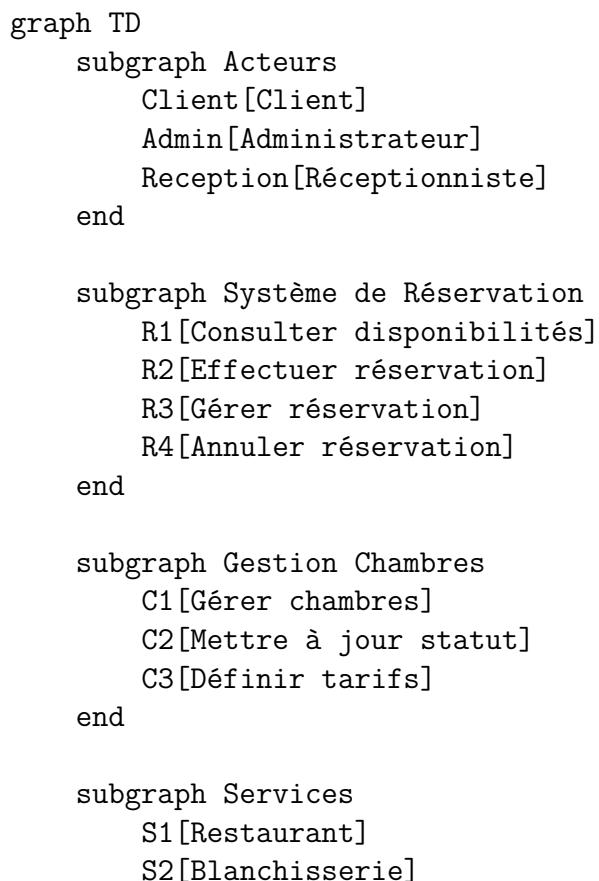
4.1.1.3 Réceptionniste

- **Rôle** : Interface entre l'hôtel et les clients
- **Responsabilités** :
 - Gérer les check-in et check-out
 - Traiter les demandes spéciales
 - Gérer les services additionnels
 - Assurer le suivi des réservations

4.1.2 1.2 Description des Cas d'Utilisation Principaux

4.1.2.1 Gestion des Réservations

1. **Consulter les disponibilités**
 - Préconditions : Accès au système
 - Flux principal :
 - Saisie des dates de séjour
 - Sélection du type de chambre
 - Vérification de la disponibilité
 - Postconditions : Affichage des options disponibles
2. **Effectuer une réservation**
 - Préconditions : Chambres disponibles
 - Flux principal :
 - Sélection de la chambre
 - Saisie des informations personnelles
 - Choix des services additionnels
 - Confirmation du paiement
 - Postconditions : Réservation confirmée
3. **Gérer une réservation existante**
 - Préconditions : Réservation active
 - Flux principal :
 - Modification des dates
 - Ajout/suppression de services
 - Mise à jour des informations
 - Postconditions : Réservation mise à jour



```
S3[Services additionnels]
end
```

```
Client --> R1
Client --> R2
Client --> R4
Client --> S1
Client --> S2
```

```
Admin --> C1
Admin --> C2
Admin --> C3
Admin --> R3
```

```
Reception --> R1
Reception --> R2
Reception --> C2
Reception --> S3
```

4.2 2. Diagramme de Classes

4.2.1 2.1 Description Détailée des Classes

4.2.1.1 Client

- **Attributs :**
 - id : Identifiant unique
 - nom : Nom du client
 - prenom : Prénom du client
 - email : Adresse email
 - telephone : Numéro de téléphone
 - adresse : Adresse postale
 - preferences : Préférences spéciales
- **Méthodes :**
 - creerReservation() : Création d'une nouvelle réservation
 - annulerReservation() : Annulation d'une réservation existante
 - modifierProfil() : Mise à jour des informations personnelles
 - consulterHistorique() : Accès à l'historique des séjours

4.2.1.2 Reservation

- **Attributs :**
 - id : Identifiant unique
 - dateArrivee : Date d'arrivée
 - dateDepart : Date de départ
 - montantTotal : Coût total du séjour
 - statut : État de la réservation
 - nombrePersonnes : Nombre de personnes
 - servicesAdditionnels : Liste des services réservés

- **Méthodes :**
 - calculerMontant() : Calcul du coût total
 - confirmerReservation() : Confirmation de la réservation
 - annuler() : Annulation avec gestion des remboursements
 - modifierDates() : Modification des dates de séjour

4.2.1.3 Chambre

- **Attributs :**
 - numero : Numéro de chambre
 - type : Type de chambre (Simple, Double, Suite)
 - prixNuit : Tarif par nuit
 - disponible : État de disponibilité
 - caracteristiques : Équipements et options
 - etage : Niveau dans l'hôtel
- **Méthodes :**
 - verifierDisponibilite() : Vérification des dates disponibles
 - mettreAJourStatut() : Mise à jour de l'état
 - calculerPrix() : Calcul du prix selon la période
 - planifierMaintenance() : Programmation de l'entretien

4.2.2 2.2 Relations entre Classes

4.2.2.1 Associations

1. **Client - Reservation :**
 - Cardinalité : 1..* (un client peut avoir plusieurs réservations)
 - Attributs d'association : dateCreation, canalReservation
2. **Reservation - Chambre :**
 - Cardinalité : *..1 (plusieurs réservations pour une chambre)
 - Attributs d'association : dureeOccupation, tarifApplique
3. **Reservation - Service :**
 - Cardinalité : .. (relation many-to-many)
 - Classe d'association : ReservationService

4.2.2.2 Hiérarchies

1. **Service (classe abstraite)**
 - Restaurant (spécialisation)
 - Blanchisserie (spécialisation)
 - SpaFitness (spécialisation)

```
classDiagram
    class Client {
        +int id
        +String nom
        +String prenom
        +String email
        +String telephone
        +creerReservation()
    }
```

```
+annulerReservation()
}

class Reservation {
    +int id
    +Date dateArrivee
    +Date dateDepart
    +double montantTotal
    +String statut
    +calculerMontant()
    +confirmerReservation()
    +annuler()
}

class Chambre {
    +int numero
    +String type
    +double prixNuit
    +boolean disponible
    +verifierDisponibilite()
    +mettreAJourStatut()
}

class Service {
    +int id
    +String nom
    +String description
    +double prix
    +commanderService()
    +annulerService()
}

class Personnel {
    +int id
    +String nom
    +String role
    +String service
    +gererReservation()
    +gererServices()
}

class Restaurant {
    +Menu menu
    +int capacite
    +creerReservation()
    +gererCommandes()
}
```

```

class Blanchisserie {
    +List services
    +double tarifs
    +traiterCommande()
    +suiviCommande()
}

Client "1" -- "*" Reservation
Reservation "*" -- "1" Chambre
Reservation "*" -- "*" Service
Personnel "1" -- "*" Service
Service <|-- Restaurant
Service <|-- Blanchisserie

```

4.3 3. Diagramme de Séquence - Processus de Réservation

4.3.1 3.1 Description des Interactions

4.3.1.1 Phase 1 : Recherche de Disponibilité

1. Client → Site Web
 - Action : Saisie des critères de recherche
 - Données : Dates, type de chambre, nombre de personnes
2. Site Web → Système Réservation
 - Action : Vérification des disponibilités
 - Traitement : Filtrage selon critères
3. Système Réservation → Base de Données
 - Action : Requête de disponibilité
 - Optimisation : Utilisation de cache Redis

4.3.1.2 Phase 2 : Processus de Réservation

1. Sélection et Configuration
 - Choix de la chambre
 - Sélection des services additionnels
 - Calcul du prix total
2. Validation et Paiement
 - Saisie des informations personnelles
 - Traitement du paiement
 - Confirmation de la réservation
3. Confirmation et Notification
 - Enregistrement en base de données
 - Envoi de confirmation par email
 - Mise à jour du planning

4.3.2 3.2 Flux de Données

```

sequenceDiagram
    participant C as Client
    participant W as Site Web
    participant R as Système Réservation
    participant D as Base de Données
    participant E as Email Service

    C->>W: Consulte disponibilités
    W->>R: Vérifie dates
    R->>D: Requête disponibilités
    D-->>R: Retourne chambres disponibles
    R-->>W: Affiche options
    W-->>C: Montre chambres disponibles

    C->>W: Sélectionne chambre
    C->>W: Ajoute services
    W->>R: Crée réservation
    R->>D: Enregistre réservation
    D-->>R: Confirme enregistrement
    R->>E: Envoie confirmation
    E->>C: Email confirmation
  
```

4.4 4. Diagramme d'État - État d'une Réservation

4.4.1 4.1 États Possibles d'une Réservation

4.4.1.1 1. En Attente

- **Description :** Réservation créée mais non confirmée
- **Conditions de transition :**
 - Vers Confirmée : Paiement reçu
 - Vers Annulée : Délai de paiement dépassé
- **Actions associées :**
 - Envoi de rappels
 - Vérification du paiement

4.4.1.2 2. Confirmée

- **Description :** Réservation validée et payée
- **Conditions de transition :**
 - Vers Check-in : Arrivée du client
 - Vers Annulée : Demande d'annulation
- **Actions associées :**
 - Préparation de la chambre
 - Planification des services

4.4.1.3 3. Check-in

- **Description :** Client arrivé et enregistré
- **Conditions de transition :**
 - Vers En Cours : Installation en chambre
- **Actions associées :**
 - Vérification des documents
 - Remise des clés

4.4.1.4 4. En Cours

- **Description :** Séjour en cours
- **Conditions de transition :**
 - Vers Check-out : Fin du séjour
- **Actions associées :**
 - Suivi des services
 - Gestion des demandes

4.4.1.5 5. Check-out

- **Description :** Fin du séjour
- **Conditions de transition :**
 - Vers Terminée : Chambre libérée
- **Actions associées :**
 - Facturation finale
 - Retour des clés

4.4.2 4.2 Transitions et Événements

stateDiagram-v2

```

[*] --> EnAttente: Création
EnAttente --> Confirmée: Paiement reçu
EnAttente --> Annulée: Délai dépassé
Confirmée --> CheckIn: Arrivée client
CheckIn --> EnCours: Occupation chambre
EnCours --> CheckOut: Départ client
CheckOut --> [*]
Annulée --> [*]

```

4.5 5. Diagramme de Composants

4.5.1 5.1 Description des Composants

4.5.1.1 Frontend

1. **Interface Utilisateur (UI)**
 - Technologies : React, Tailwind CSS
 - Responsabilités :
 - Affichage des pages
 - Gestion des formulaires

- Interactions utilisateur
 - Responsive design
- 2. State Management**
- Technologies : Redux, Context API
 - Responsabilités :
 - Gestion de l'état global
 - Cache local
 - Synchronisation des données

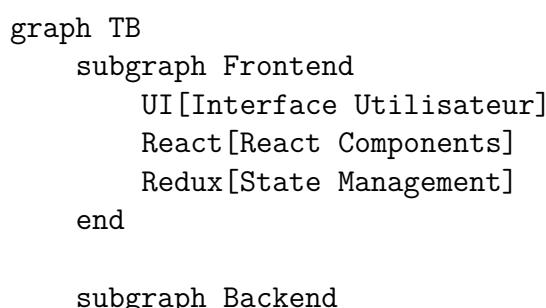
4.5.1.2 Backend

- 1. API REST**
- Technologies : Express.js
 - Endpoints principaux :
 - /api/reservations
 - /api/chambres
 - /api/services
 - /api/utilisateurs
- 2. Authentication**
- Technologies : JWT, OAuth
 - Fonctionnalités :
 - Gestion des sessions
 - Contrôle d'accès
 - Sécurité des données

4.5.1.3 Base de Données

- 1. MongoDB**
- Collections :
 - utilisateurs
 - reservations
 - chambres
 - services
- 2. Redis Cache**
- Utilisation :
 - Cache de session
 - Données temporaires
 - Files d'attente

4.5.2 5.2 Architecture Technique



```

    API [API REST]
    Auth [Authentication]
    BL [Business Logic]
end

subgraph Database
    MongoDB [MongoDB]
    Cache [Redis Cache]
end

subgraph Services
    Email [Email Service]
    Payment [Payment Gateway]
    Storage [File Storage]
end

UI --> React
React --> Redux
Redux --> API
API --> Auth
API --> BL
BL --> MongoDB
BL --> Cache
BL --> Email
BL --> Payment
BL --> Storage

```

4.6 6. Diagramme de Déploiement

4.6.1 6.1 Infrastructure Physique

4.6.1.1 Environnement Client

1. Navigateur Web
 - Support des technologies modernes
 - Progressive Web App (PWA)
 - Service Workers pour le mode hors ligne
2. Application Mobile
 - Version iOS et Android
 - Framework React Native
 - Synchronisation locale

4.6.1.2 Serveurs Frontend

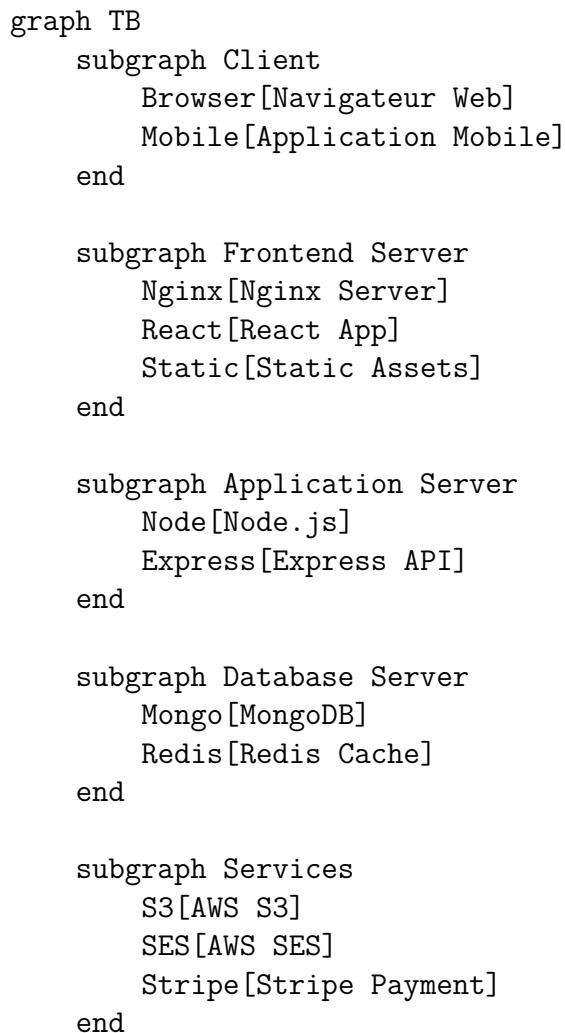
1. Nginx Server
 - Configuration :
 - Load balancing
 - SSL/TLS
 - Compression gzip

- Cache statique
- 2. **CDN**
 - Distribution globale
 - Cache des assets
 - Optimisation des performances

4.6.1.3 Serveurs Backend

- 1. **Application Server**
 - Spécifications :
 - Node.js v18+
 - 4 CPU cores
 - 16GB RAM
 - Scaling automatique
- 2. **Database Server**
 - Configuration :
 - MongoDB cluster
 - Réplication
 - Backup automatique

4.6.2 6.2 Infrastructure Cloud



Browser --> Nginx
Mobile --> Nginx
Nginx --> React
Nginx --> Static
React --> Express
Express --> Mongo
Express --> Redis
Express --> S3
Express --> SES
Express --> Stripe