

Rapport de Projet : Gestion des Hôtels

Introduction:

L'objectif de ce projet est de concevoir une application web pour la gestion de huit hôtels. Cette application permettra de gérer les réservations, les chambres, les consommations, les factures, ainsi que l'archivage des données. Ce rapport présente l'analyse du système, les diagrammes UML, et les modalités de conception.

Analyse des besoins:

Caractéristiques principales

1. Chaque hôtel est caractérisé par :

- Nom
- Adresse
- Téléphone
- Classe (1 à 5 étoiles)
- Code unique(ID_hotel)

2. Les chambres sont caractérisées par :

- Numéro
- Numéro de téléphone
- Catégorie

3. Les catégories des chambres :

- Code unique
- Description

4. Les réservations :

- N° de réservation
- Date de debut
- date de fin
- Montant des arrhes

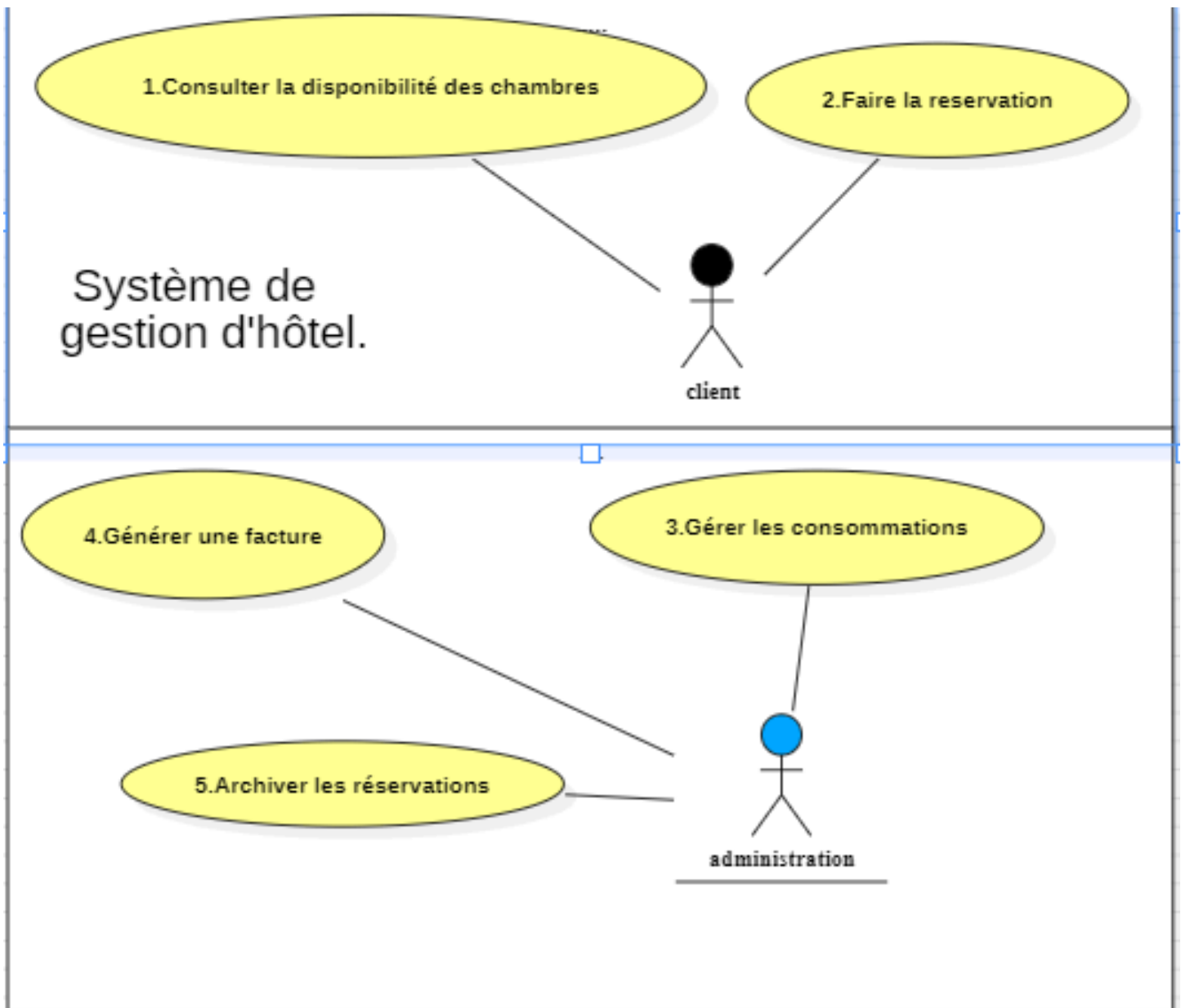
5. Les consommations :

- N° de consommation
- Date et heure
- Détails des prestations consommées

6. Les clients :

- Identité complète
- Coordonnées

1. Diagramme de cas d'utilisation



Ce diagramme montre les principales interactions entre les utilisateurs et le système

Acteurs principaux :

- Client

- Administrateur

Cas d'utilisation :

- Consultation des disponibilités des chambres
- faire la reservation
- Gestion des consommations
- Génération de factures
- Archivage des réservations

2.Diagramme de classes:

Classes principales :

- **Hotel**
- **Chambre**
- **Catégorie**
- **Client**
- **Réservation**
- **Consommation**

Relations :

- Un hôtel contient plusieurs chambres
- Une chambre appartient à une catégorie
- Une réservation est associée à un client et une chambre
- Une facture regroupe une réservation et les consommations

Méthodes :

Hotel:

- **ajouterCategorie(Categorie c)** : Permet d'ajouter une catégorie à l'hôtel
- **supprimerCategorie(Categorie c)** : Supprime une catégorie de l'hôtel
- **obtenirCategories()** : Retourne la liste des catégories disponibles dans l'hôtel
- **consulterDisponibilite()** : Vérifie la disponibilité des chambres de l'hôte

categorie:

- **modifierPrix(nouveauPrix)** : Permet de mettre à jour le prix d'une

catégorie de chambre

chambre:

- **reserver()** : Effectue la réservation de la chambre si elle est disponible
- **liberer()** : Libère la chambre lorsque le client quitte

Client:

- **reserverChambre()** : Permet au client de faire une réservation en fonction de ses préférences
- **annulerReservation()** : Permet au client d'annuler une réservation existante

Reservation

- **enregistrer()** : Enregistre la réservation dans le système.
- **archiver()** : Archive la réservation une fois le séjour terminé (réservations effectuées il y a plus de 6 mois)
- **calculerTotal()** : Calcule le total de la facture, y compris les consommations et les services supplémentaires

Consommation:

- **enregistrerConsommation()** : Enregistre une nouvelle consommation liée à une réservation.
- **obtenirDetails()** : Retourne les détails d'une consommation (montant, date,..)

Facture:

calculerMontantTotal() :

- Cette méthode pourrait calculer le montant total de la facture en additionnant :
 - Le prix de la chambre (en fonction de la catégorie de la chambre et de l'hôtel).
 - Le montant total des consommations (par exemple, le coût du petit déjeuner, du déjeuner, etc.).
 - Autres frais éventuels (par exemple, services supplémentaires).

ajouterConsommation(consommation: Consommation) :

- Cette méthode permet d'ajouter une consommation spécifique à la facture, de manière à suivre l'évolution du montant à payer par le client. Elle prend en paramètre un objet de type Consommation et met à jour le montant de la facture.

genererFacture() :

- Cette méthode finalise la facture. Elle calcule le montant total à partir des chambres et des consommations associées, et elle marque la facture comme étant prête à être réglée.

afficherFacture() :

- Cette méthode pourrait afficher la facture (en texte, ou sous forme de fichier PDF par exemple) avec tous les détails : prix de la chambre, consommations, montant total, et informations sur le client et la réservation.

RESUME :

L'objectif de cette application est de permettre la gestion efficace de huit hôtels, en incluant des fonctionnalités telles que la réservation de chambres, la gestion des consommations pendant le séjour des clients, l'archivage des réservations passées, la génération de factures, et le suivi des prestations proposées par chaque hôtel. Chaque hôtel est défini par plusieurs attributs comme son nom, son adresse, son téléphone, sa classe (étoiles), et possède un ensemble de chambres organisées par catégories. La gestion des réservations comprend également l'enregistrement des informations des clients et la possibilité de consulter la disponibilité des chambres en fonction des dates.

Objectifs et fonctionnalités principales :

- Consultation de la disponibilité des chambres : Permet au client de connaître les chambres disponibles à une date donnée.
- Enregistrement de la réservation : Permet au client de réserver une chambre selon sa catégorie, la période de séjour et la classe de l'hôtel.

- Gestion des consommations : Permet d'enregistrer les consommations du client pendant son séjour (repas, services, etc.).
- Archivage des réservations : Les réservations effectuées depuis plus de 6 mois sont archivées pour une gestion à long terme.
- Génération de la facture : À la fin du séjour, la facture est générée, incluant le prix de la chambre et les prestations consommées.

Utilité de construire des diagrammes de classes et des diagrammes de cas d'utilisation pour cette application

1. Diagramme de classes

Le diagramme de classes est un outil fondamental dans la phase de conception d'une application orientée objet. Il permet de modéliser les différentes entités du système et les relations entre elles. Pour l'application de gestion d'hôtels, il est crucial de représenter les objets principaux comme Hôtel, Chambre, Réservation, Consommation, Facture, Prestation, et Client, ainsi que leurs attributs et méthodes.

Utilité du diagramme de classes :

- Clarté sur les relations : un hôtel possède plusieurs chambres, une chambre appartient à une catégorie, et chaque réservation est liée à un client et à une chambre.
- Modélisation des interactions : Le diagramme aide à déterminer comment les objets interagissent entre eux, ce qui est essentiel pour le développement des fonctionnalités comme la gestion des réservations et la génération des factures.
- Structure logique : Il aide à structurer l'application en différentes classes cohérentes, chacune ayant des responsabilités bien définies.
- Facilite le développement : En définissant clairement les classes, leurs attributs et leurs méthodes, le diagramme de classes sert de base pour l'implémentation du code et évite les erreurs lors de la programmation.
- Maintenance future : Il facilite la compréhension de l'architecture du système pour toute modification ou ajout de fonctionnalités à l'avenir.

2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de visualiser les interactions entre les utilisateurs (acteurs) et le système. Il représente les fonctionnalités principales du système sous forme de cas d'utilisation, décrivant ce que l'utilisateur peut accomplir avec l'application. Ce diagramme est essentiel pour déterminer les exigences fonctionnelles de l'application.

Utilité du diagramme de cas d'utilisation :

- Identification des besoins utilisateurs : Il aide à identifier les besoins des utilisateurs (clients, personnel de l'hôtel, ...) en montrant les tâches qu'ils peuvent accomplir. Par exemple, un client peut réserver une chambre, consulter la disponibilité, ou générer une facture.
- Clarté des fonctionnalités : Il clarifie les actions principales de l'application (réservation, paiement, consultation des consommations, etc.) et aide à les organiser.
- Facilite la communication : Le diagramme sert de référence pour la communication entre les parties prenantes (développeurs, clients, responsables d'hôtels) pour s'assurer que les besoins fonctionnels sont bien compris et pris en compte.
- Base pour les tests : Il permet de définir des scénarios de tests en fonction des cas d'utilisation, afin de garantir que toutes les fonctionnalités sont bien implémentées.
- Préparation à l'implémentation : En définissant clairement les interactions et les processus utilisateur, il prépare l'équipe de développement à la mise en œuvre des fonctionnalités de l'application.

Conclusion:

La construction des diagrammes de classes et des cas d'utilisation est essentielle pour la conception de l'application de gestion d'hôtels. Le diagramme de classes permet de visualiser les entités du système (comme **Hôtel**, **Chambre**, **Réservation**, **Consommation**, **Facture**, etc.) ainsi que leurs relations et attributs, ce qui facilite l'implémentation du

système et assure une gestion efficace des hôtels et des réservations. Le diagramme de cas d'utilisation, quant à lui, identifie les principales interactions entre les utilisateurs (clients, employés de l'hôtel, système) et le système lui-même, en mettant l'accent sur les fonctionnalités nécessaires telles que la réservation, l'archivage des réservations, la gestion des consommations et la génération de factures.

En plus de ces diagrammes, d'autres diagrammes UML peuvent être ajoutés pour enrichir la modélisation du système et faciliter sa compréhension et son développement :

1. **Diagrammes de séquence** : lorsqu'un client effectue une réservation, un diagramme de séquence peut illustrer les étapes de l'interaction entre le client, le système de réservation, et l'hôtel pour valider la disponibilité des chambres, enregistrer la réservation, et confirmer le paiement des arrhes.
2. **Diagrammes d'activité** : le processus de gestion des réservations peut être décrit à l'aide d'un diagramme d'activité, montrant l'enchaînement des tâches depuis la demande de réservation jusqu'à l'enregistrement des consommations et la génération de la facture. Un autre exemple pourrait être le processus de gestion des consommations pendant le séjour, permettant de suivre les services utilisés par le client.
3. **Diagrammes d'états-transitions** : l'état d'une **réservation** pourrait passer par des étapes comme "Réservation en attente", "Réservation confirmée", "Réservation annulée" ou "Séjour terminé". Ce diagramme permettrait de suivre le cycle de vie d'une réservation et de son paiement, en indiquant les transitions entre les états selon les actions de l'utilisateur.
4. **Diagrammes de composants** : Un diagramme de composants peut être utilisé pour définir l'architecture technique de l'application. Il permet de visualiser les différents modules logiciels du système, comme la gestion des hôtels, des réservations, des consommations et des factures, et de montrer comment ces composants interagissent les uns avec les autres.

Ces diagrammes supplémentaires permettent de mieux comprendre les processus métier complexes et de garantir que l'application de gestion des hôtels sera non seulement fonctionnelle, mais aussi évolutive et maintenable à long terme. En combinant ces diagrammes avec ceux des classes et des cas d'utilisation, on obtient une vue complète du système, ce qui facilite le développement, la gestion des exigences, et la communication entre les parties prenantes du projet