



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de
HONORIS UNITED UNIVERSITIES

RAPPORT

DE PROJET

SYSTÈME DE GESTION HÔTELIÈRE (HMS)

Réalisé par :

- Belgas Zainab

Encadré par :

- Pr. ABDERRAHIM LARHLIMI

INTITULÉ DU MODULE : JAVA AVANCÉ

Année Universitaire : 2025-2026

2. Remerciements

Nous tenons à exprimer notre profonde gratitude à notre encadrant pédagogique, **Pr. Abderrahim Larhlimi**, pour son accompagnement rigoureux, sa disponibilité constante et la qualité de ses orientations tout au long de la réalisation de ce projet. Ses conseils méthodologiques, son sens critique et son exigence académique ont constitué un véritable moteur pour la bonne conduite de ce travail. Grâce à son encadrement, nous avons pu consolider nos connaissances en Java Avancé, approfondir notre compréhension des architectures logicielles et adopter une démarche professionnelle dans la conception et l'implémentation d'un système informatique.

Nous adressons également nos sincères remerciements à l'**administration de l'EMSI** ainsi qu'à l'ensemble du **corps professoral**, pour les moyens techniques, pédagogiques et logistiques mis à notre disposition. L'environnement de travail offert par l'établissement, la qualité des enseignements et l'accès aux ressources nécessaires ont grandement contribué à la réussite de ce projet académique.

Nous souhaitons aussi remercier nos enseignants des différents modules de la formation 4IIR, dont les cours complémentaires ont enrichi notre vision globale de l'ingénierie logicielle et nous ont permis d'aborder ce projet avec une approche structurée, cohérente et conforme aux standards professionnels.

Enfin, nous tenons à exprimer notre reconnaissance à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet, que ce soit par leurs encouragements, leurs conseils ou leur soutien moral. Ce travail représente non seulement l'aboutissement d'un apprentissage académique, mais également une étape importante dans notre parcours de formation en tant que futurs ingénieurs informaticiens.

3. Table des Matières

1. Introduction Générale.....	
2. Analyse et Conception.....	
3. Environnement Technique.....	
4. Architecture et Implémentation.....	
5. Interface Utilisateur et Tests.....	
6. Conclusion et Perspectives.....	
7. Webographie.....	

4. Introduction Générale

Présentation Générale du Projet

Le présent rapport s'inscrit dans le cadre du module **Java Avancé** dispensé aux étudiants de la filière **4IIR – Génie Informatique** à l'EMSI. Il vise à présenter de manière détaillée la conception, le développement et la mise en œuvre d'un **Système de Gestion Hôtelière (Hotel Management System – HMS)**.

L'industrie hôtelière est aujourd'hui confrontée à une forte concurrence et à des exigences accrues en matière de qualité de service, de rapidité de traitement et de fiabilité des informations. La transformation numérique est devenue un levier stratégique incontournable pour améliorer l'expérience client et optimiser la gestion interne des établissements hôteliers.

Dans ce contexte, le développement d'un système informatique intégré permet de centraliser les données, d'automatiser les processus critiques et de réduire considérablement les erreurs humaines.

4.1 Contexte du Projet

Le secteur hôtelier moderne repose sur des systèmes d'information performants afin de garantir une gestion efficace des chambres, des clients, des réservations et de la facturation. Les méthodes traditionnelles de gestion (papier ou fichiers Excel) montrent rapidement leurs limites face aux exigences de disponibilité en temps réel, de fiabilité des données et de qualité de service.

Ce projet s'inscrit dans le cadre du module **Java Avancé** de la formation 4IIR à l'EMSI. Il vise à concevoir et développer une application de gestion hôtelière basée sur les principes de la programmation orientée objet et des architectures logicielles modernes.

4.2 Problématique

Comment concevoir un système logiciel capable de centraliser la gestion des chambres, des clients, des factures et des services hôteliers, tout en garantissant la cohérence des données, la sécurité des accès et une interface utilisateur intuitive ?

4.3 Objectifs du Projet

Les principaux objectifs du projet sont :

- Gérer le cycle de vie complet des clients (check-in / check-out).
- Assurer la gestion des chambres et de leur disponibilité.
- Automatiser la génération et le calcul des factures.
- Gérer les services additionnels (commandes de nourriture, inventaire).
- Mettre en place une architecture logicielle robuste et évolutive.

5. Analyse et Conception

Cette partie constitue la phase fondamentale du projet. Elle permet d'analyser le problème à résoudre, d'identifier les besoins des utilisateurs et de concevoir une solution logicielle cohérente avant toute phase d'implémentation.

5.1 Spécification des Besoins

5.1.1 Besoins Fonctionnels

ID	Description
BF01	Le système doit permettre la gestion des chambres (ajout, modification, suppression).
BF02	Le système doit permettre l'enregistrement et la gestion des clients.
BF03	Le système doit attribuer une chambre à un client.
BF04	Le système doit générer et gérer les factures des clients.
BF05	Le système doit gérer les commandes de nourriture et services.
BF06	Le système doit gérer l'inventaire de l'hôtel.

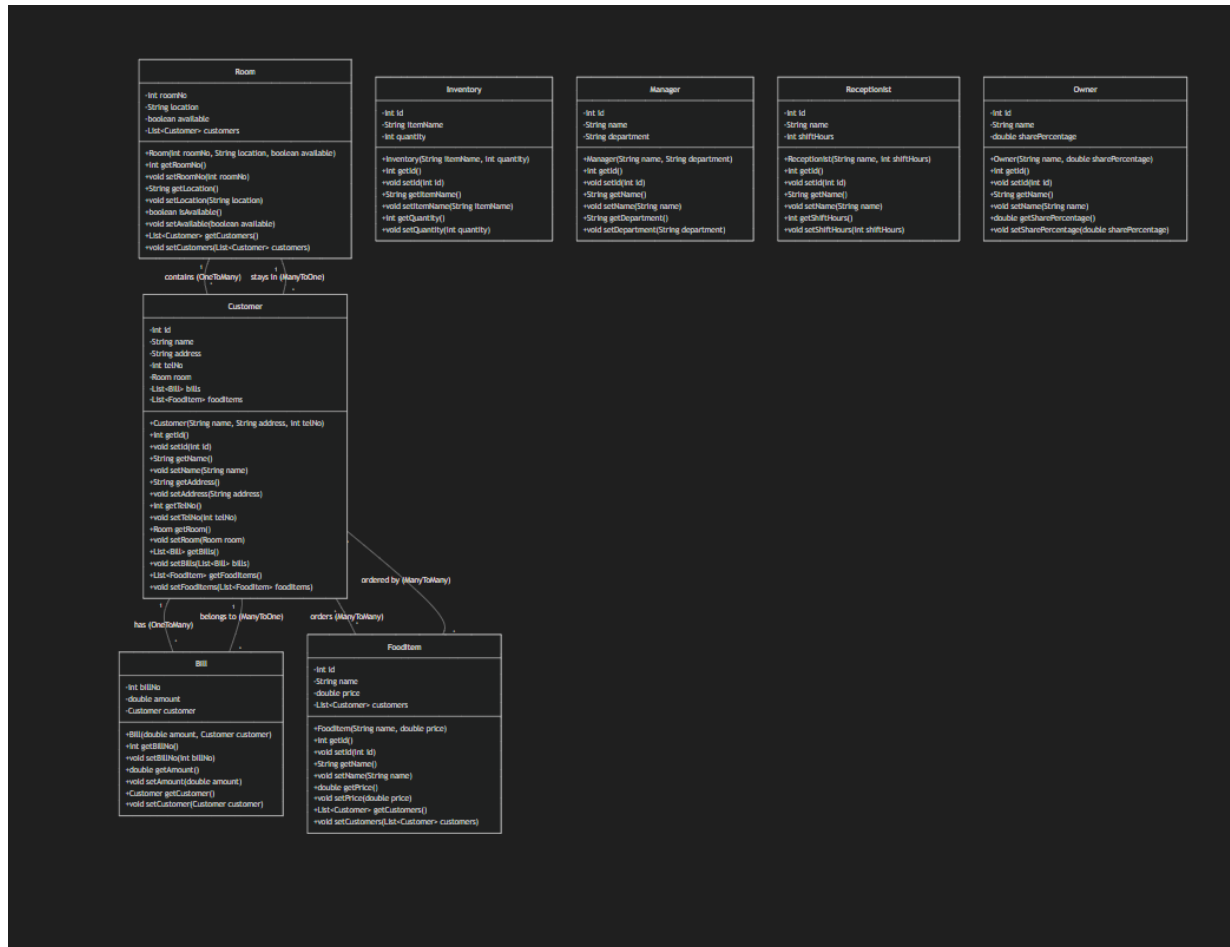
5.1.2 Besoins Non-Fonctionnels

ID	Description
BNF01	L'application doit respecter les principes de la programmation orientée objet.
BNF02	Le code doit être modulaire et maintenable.
BNF03	Le système doit être extensible pour intégrer de nouvelles fonctionnalités.
BNF04	Les données doivent être cohérentes et persistantes.

5.2 Conception UML

5.2.1 Diagramme de Classes

Le diagramme de classes constitue le cœur de la conception du projet. Il représente les entités principales du système ainsi que leurs relations.



Les principales classes sont :

- **Room** : représente une chambre de l'hôtel.
- **Customer** : représente un client.
- **Bill** : représente une facture associée à un client.
- **FoodItem** : représente un élément de restauration commandé.
- **Inventory** : représente les ressources matérielles disponibles.
- **Manager, Receptionist, Owner** : représentent les différents rôles du personnel.

Relations principales :

- Un **Customer** peut avoir plusieurs **Bill** (1..*).
- Une **Room** peut accueillir plusieurs **Customer** dans le temps.
- Un **Customer** peut commander plusieurs **FoodItem** (relation Many-to-Many).

5.3 Conception de la Base de Données

5.3.1 Modèle Logique de Données (MLD)

Le projet adopte une approche orientée objet, où chaque classe métier correspond logiquement à une table en base de données.

Tables principales :

- ROOM (room_no, location, available)
- CUSTOMER (id, name, address, telNo)
- BILL (billNo, amount, customer_id)
- FOOD_ITEM (id, name, price)
- INVENTORY (id, itemName, quantity)

5.3.2 Dictionnaire de Données (Extrait)

Champ	Type	Contrainte
room_no	INT	PK
available	BOOLEAN	NOT NULL
name	VARCHAR	NOT NULL
amount	DOUBLE	>= 0

6. Environnement Technique

Cette partie présente l'ensemble des choix technologiques effectués pour la réalisation du projet, ainsi que les justifications associées à ces choix.

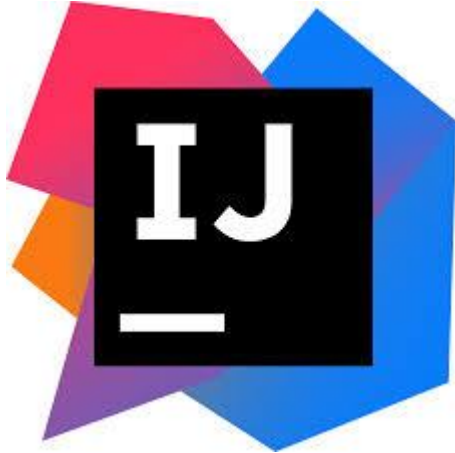
6.1 Langage de Programmation

- **Java 17 (LTS)** : utilisé pour le développement backend et la logique métier.



6.2 Environnement de Développement

- **IDE** : IntelliJ IDEA



- **Système de versionnement** : Git



6.3 Gestion de Projet

- **Apache Maven** : gestion des dépendances et automatisation du cycle de build.



6.4 Base de Données

- **SGBD** : H2 Database (mode développement).
- **ORM** : Hibernate (JPA) pour la persistance des données.

7. Architecture et Implémentation

Cette section décrit en profondeur l'architecture logicielle adoptée, l'organisation du code source et les mécanismes techniques mis en œuvre pour assurer la robustesse, la maintenabilité et l'évolutivité de l'application.

7.1 Architecture Logicielle

Le projet suit une architecture en couches :

- **entity** : classes métier (Room, Customer...).
- **dao** : accès aux données.
- **service** : logique métier.
- **util** : classes utilitaires.

7.2 Design Patterns Utilisés

- **Singleton** : utilisé pour la gestion de la connexion à la base de données.
- **DAO (Data Access Object)** : pour séparer la logique métier de l'accès aux données.

7.3 Extraits de Code Clés

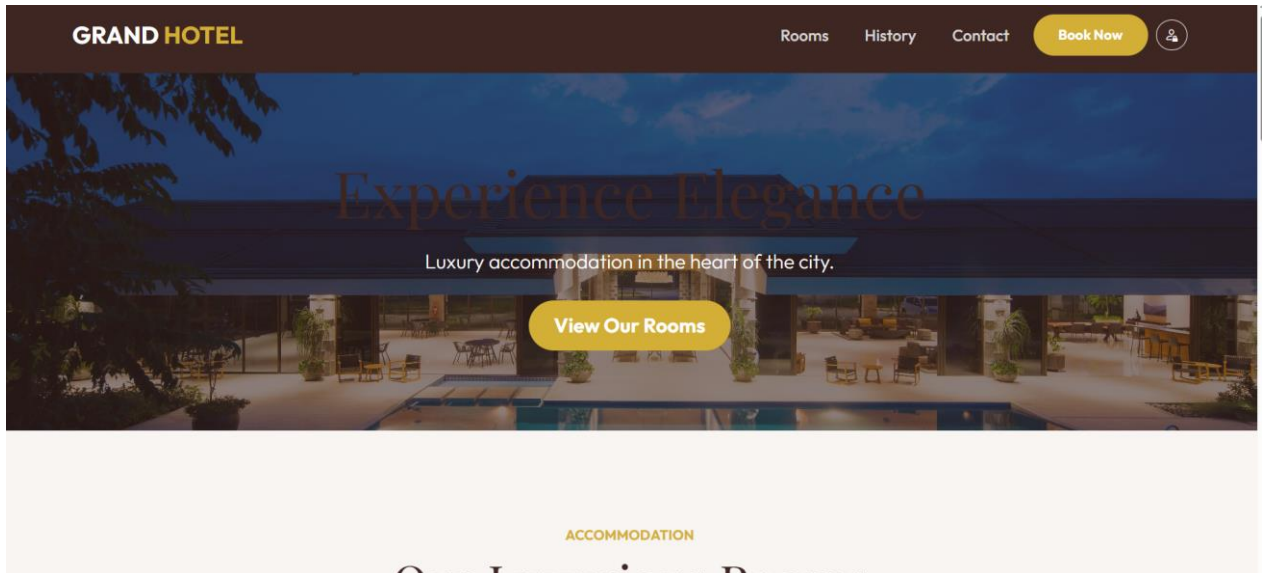
- Gestion générique des opérations CRUD via un DAO générique.
 - Configuration Hibernate pour la persistance automatique des entités.
-

8. Interface Utilisateur et Tests

Cette partie est consacrée à la présentation des interfaces graphiques de l'application ainsi qu'aux différents scénarios de tests réalisés afin de valider le bon fonctionnement du système.

8.1 Présentation des Interfaces

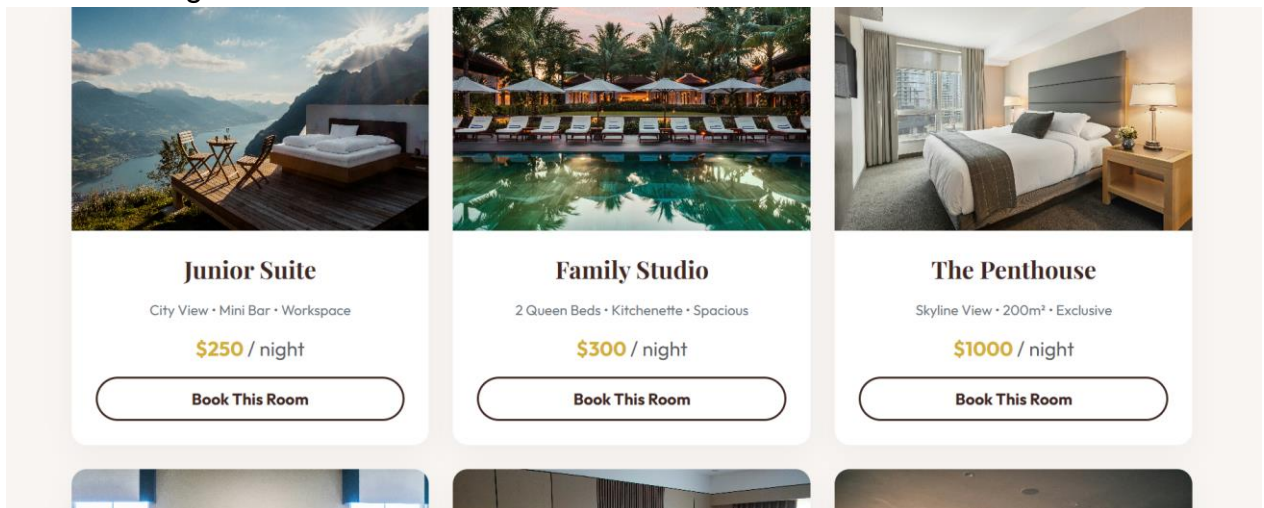
- Interface de site



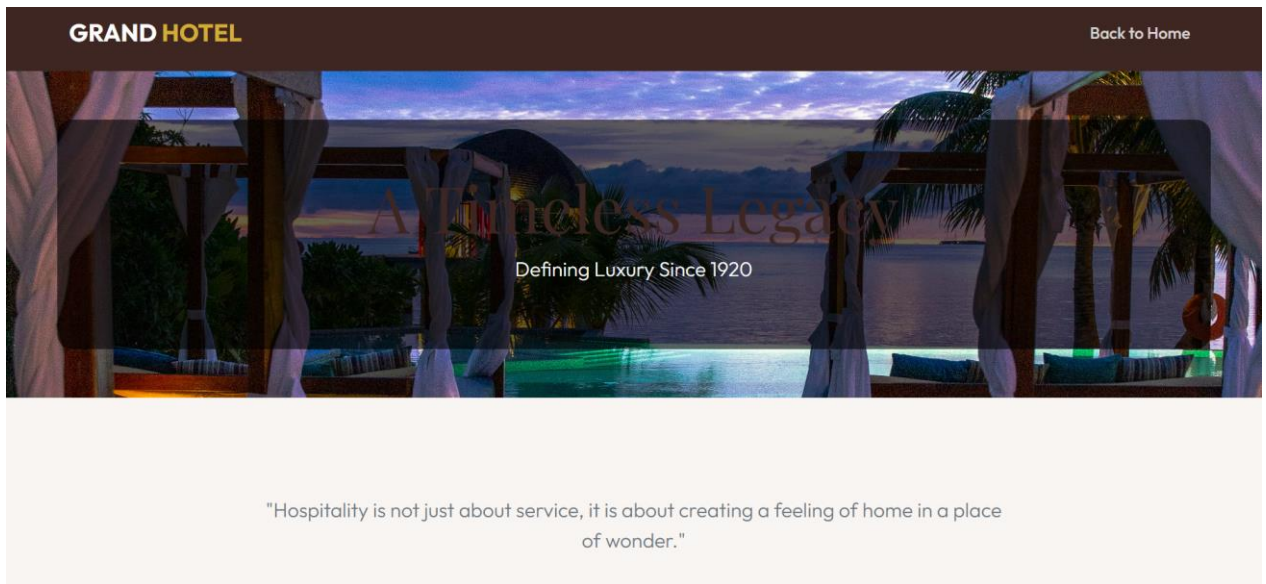
- Interface de recherche de chambre



- Interface de gestion des chambres.



- Interface de historie



- Interface de réservation

GRAND HOTEL Rooms History Contact Book Now 👤

Deluxe Suite
Ocean View • King Bed • Breakfast
\$350 / night
[Book This Room](#)

Complete Reservation ✕

ROOM SELECTED
Deluxe Suite

FIRST NAME LAST NAME

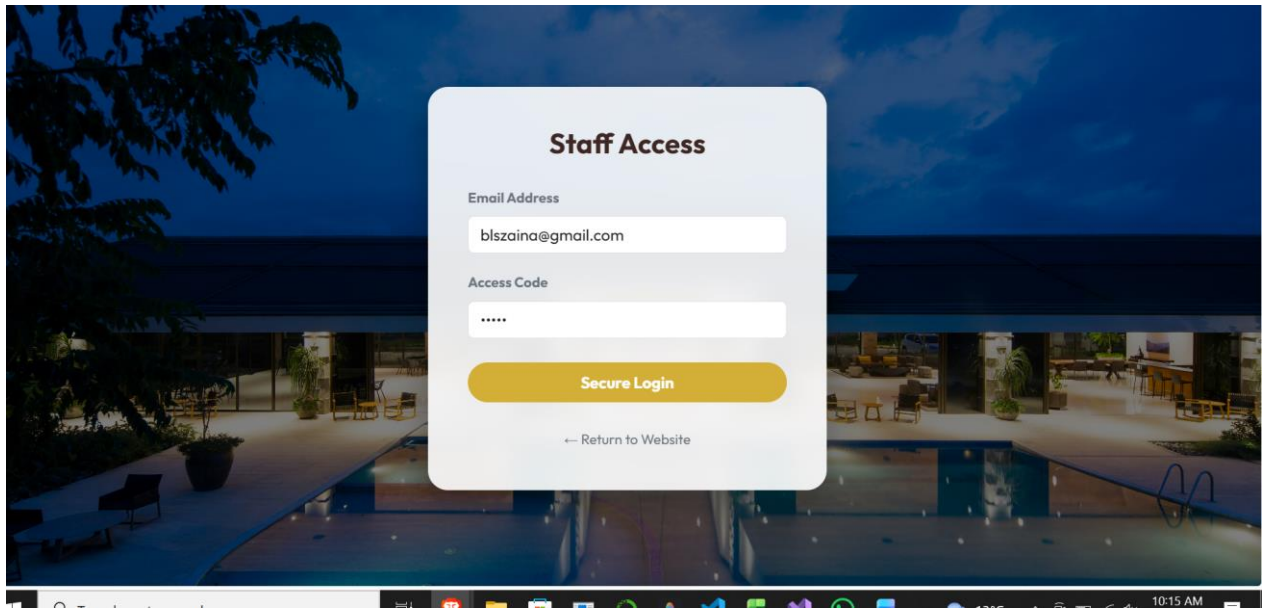
CHECK-IN CHECK-OUT

① 39 nights x \$350
Total Estimated: **\$13650**

[Confirm Reservation](#)

Presidential Suite
Top Floor • Private Pool • Butler
\$850 / night
[Book This Room](#)

- Partie login de ADMIN



- Interface de gestion les reservations.

- Reservations
- Customers
- Rooms
- Logout

Management Console

System Live

Incoming Reservations

Refresh

Ref	Guest Name	Room Type	Dates	Total	Status	Actions
#3058	SALAH swileh	Single Economy	2026-01-09 (7 nights)	\$1050	Confirmed	
#1447	hajar belgas	Honeymoon Suite	2030-12-27 (14 nights)	\$7000	Confirmed	
#1883	Abdallah Belgas	Deluxe Suite	2026-01-28 (7 nights)	\$2450	Confirmed	
#3344	Chaimaa Zerrag	Presidential Suite	2027-01-13 (16 nights)	\$13600	Confirmed	
#1145	ilham afouaart	Family Studio	2026-01-21 (14 nights)	\$4200	Pending	
#2327	khadija jaafari	Deluxe Suite	2028-06-15 (39 nights)	\$13650	Pending	

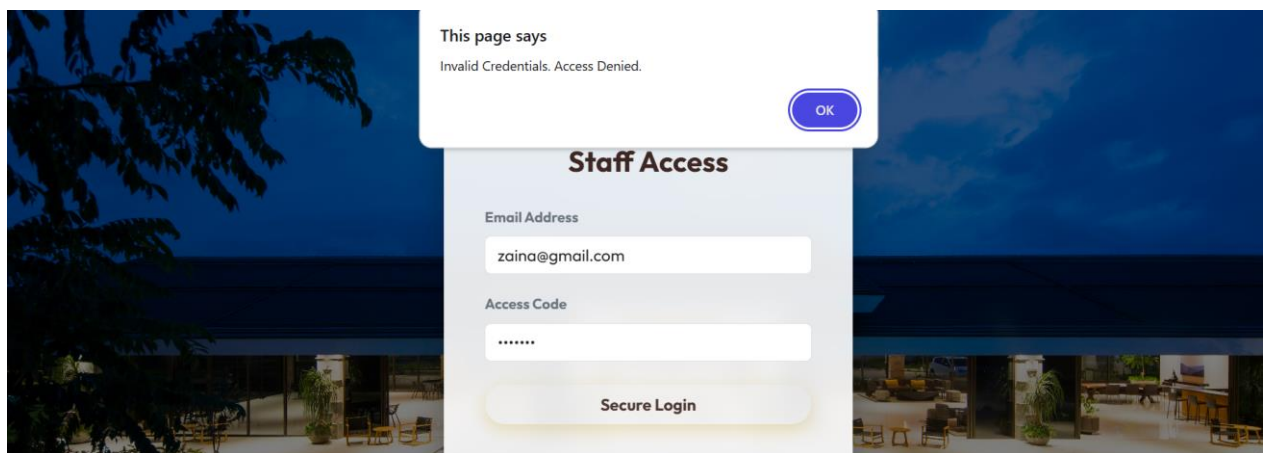
8.2 Scénarios de Test

Tests Nominaux

- Ajout d'un client avec succès.

Ref	Guest Name	Room Type	Dates	Total	Status	Actions
#3058	SALAH swileh	Single Economy	2026-01-09 (7 nights)	\$1050	Confirmed	<input type="checkbox"/> <input type="checkbox"/>
#1447	hajar belgas	Honeymoon Suite	2030-12-27 (14 nights)	\$7000	Confirmed	<input type="checkbox"/> <input type="checkbox"/>
#1883	Abdallah Belgas	Deluxe Suite	2026-01-28 (7 nights)	\$2450	Confirmed	<input type="checkbox"/> <input type="checkbox"/>
#3344	Chaimaa Zerrag	Presidential Suite	2027-01-13 (16 nights)	\$13600	Confirmed	<input type="checkbox"/> <input type="checkbox"/>
#1145	Ilham afouaart	Family Studio	2026-01-21 (14 nights)	\$4200	Pending	<input type="checkbox"/> <input type="checkbox"/>
#2327	khadija jaafari	Deluxe Suite	2028-06-15 (39 nights)	\$13650	Pending	<input type="checkbox"/> <input type="checkbox"/>

- Attribution d'une chambre disponible.
- Tests d'Erreurs



- Tentative d'attribution d'une chambre indisponible.

#2327	khadija jaafari	Deluxe Suite	2028-06-15 (39 nights)	\$13650	Pending	<input type="checkbox"/> <input type="checkbox"/>
-------	-----------------	--------------	------------------------	---------	---------	---------------------------------------------------

9. Conclusion et Perspectives

Cette dernière partie dresse un bilan global du projet, tant sur le plan technique que pédagogique, et propose des pistes d'amélioration pour des évolutions futures.

9.1 Bilan Technique

Ce projet a permis de développer une application complète de gestion hôtelière en appliquant les concepts avancés de Java et de la programmation orientée objet.

9.2 Bilan Personnel

La réalisation de ce projet m'a permis de renforcer mes compétences en Java Avancé, en conception UML et en architecture logicielle.

9.3 Difficultés Rencontrées

- Gestion des relations entre classes.
- Organisation de l'architecture en couches.

9.4 Perspectives d'Amélioration

- Ajout d'une interface Web (Spring Boot).
 - Intégration du paiement en ligne.
 - Génération des factures en PDF.
-

10. Webographie

- Documentation Java : <https://docs.oracle.com>
- Documentation Hibernate : <https://hibernate.org>
- Cours Java Avancé – EMSI