Hotel Reservation System

Project Report  
  
IF3E - Database Systems Project  
Université de Technologie de Belfort-Montbéliard (UTBM)  
  
Authors:  
STRICH Gauthier

RASQUIN Arnaud

MATHIEU Antoine

# Table of Contents

**Table of Contents**

1. **Introduction** ………………………………………………… 1  
   1.1. Context and Objectives ……………………………………… 1  
   1.2. Scope of the Project ………………………………………… 2  
   1.3. Methodology …………………………………………………… 2
2. **System Analysis** ……………………………………………… 3  
   2.1. Requirement Analysis ………………………………………… 3  
   2.2. Functional and Non-Functional Requirements ……………… 4  
   2.3. Use Case Diagrams …………………………………………… 5
3. **System Design** ………………………………………………… 6  
   3.1. Conceptual Data Model (CDM) ……………………………… 6  
   3.2. Logical Data Model …………………………………………… 7  
   3.3. System Architecture …………………………………………… 8  
   3.4. Interface Design ……………………………………………… 9
4. **Implementation** ……………………………………………… 10  
   4.1. Tools and Technologies ……………………………………… 10  
   4.2. Database Implementation …………………………………… 11  
   4.3. Application Modules ………………………………………… 12  
   4.4. Security Measures …………………………………………… 13
5. **Testing and Validation** …………………………………… 14  
   5.1. Test Plan ……………………………………………………… 14  
   5.2. Test Cases and Results ……………………………………… 15  
   5.3. User Feedback ………………………………………………… 16
6. **Conclusion and Perspectives** …………………………… 17
7. **References** …………………………………………………… 18
8. **Appendices** …………………………………………………… 19  
   8.1. Screenshots of the System ………………………………… 19  
   8.2. Code Snippets ………………………………………………… 20  
   8.3. Additional Diagrams ………………………………………… 21

# 1. Introduction

The Hotel Reservation System project aims to design and implement a relational database and web application for managing hotel operations such as reservations, guest information, room management, billing, and feedback. The goal is to streamline booking processes, ensure real-time room availability, and improve guest satisfaction through an integrated and user-friendly system.

[Insert introductory image or project logo here]

# 2. Objectives and Scope

The project covers key hotel management processes including:  
- Guest management  
- Room management  
- Reservation handling  
- Billing and payments  
- Customer feedback and reports

- feedback management

- facilities management and options

The system provides functionalities for multiple roles (guests, receptionists, managers, administrators) and ensures data integrity and usability.

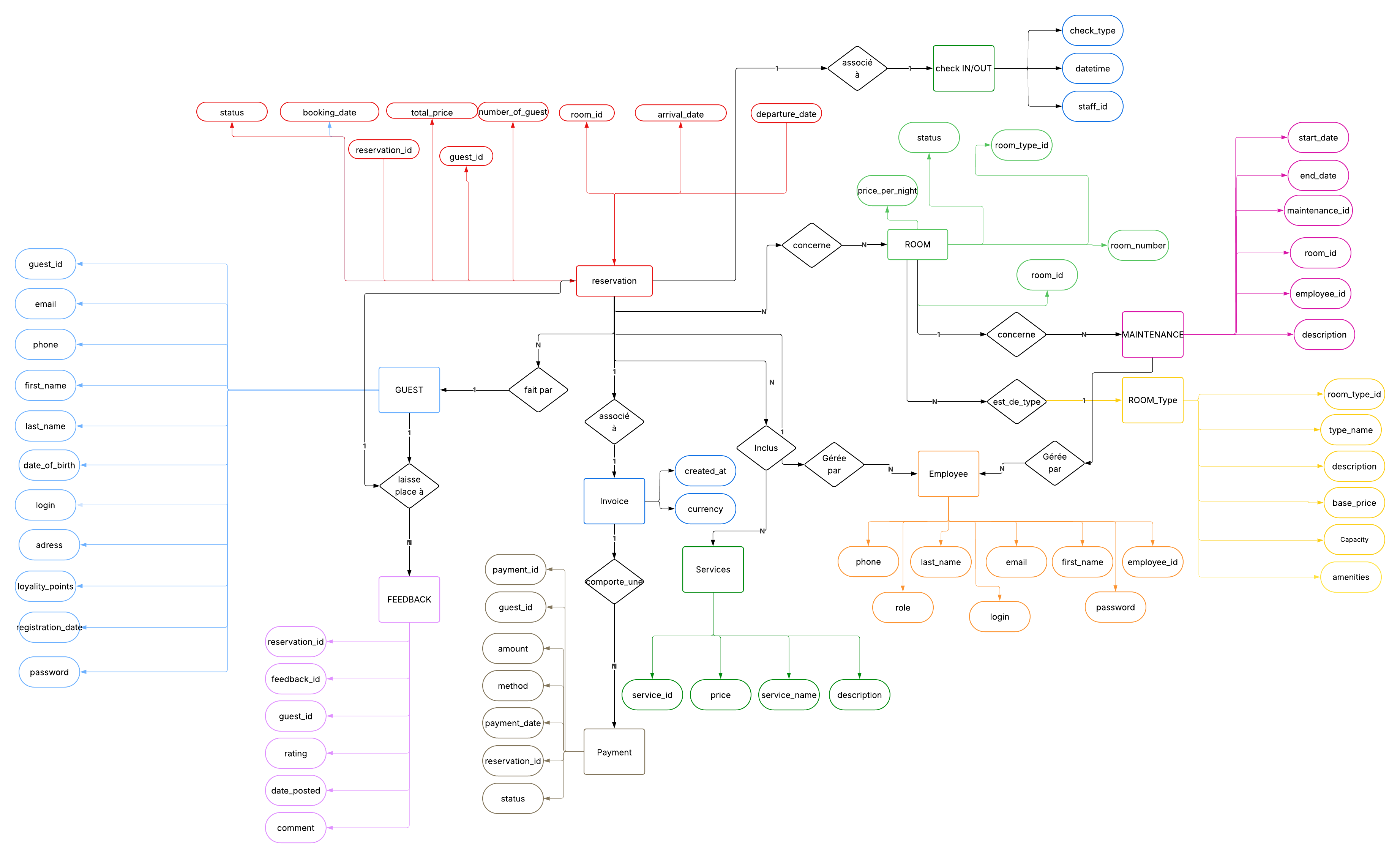
# 3. Methodology

The design process followed the main steps of information system development:  
1. Conceptual Data Model (Entity-Relationship Diagram)  
2. Relational Database Model Implementation  
3. Development of SQL queries and PHP web interfaces  
4. Integration of functionalities for multiple users  
5. Testing and validation of the system

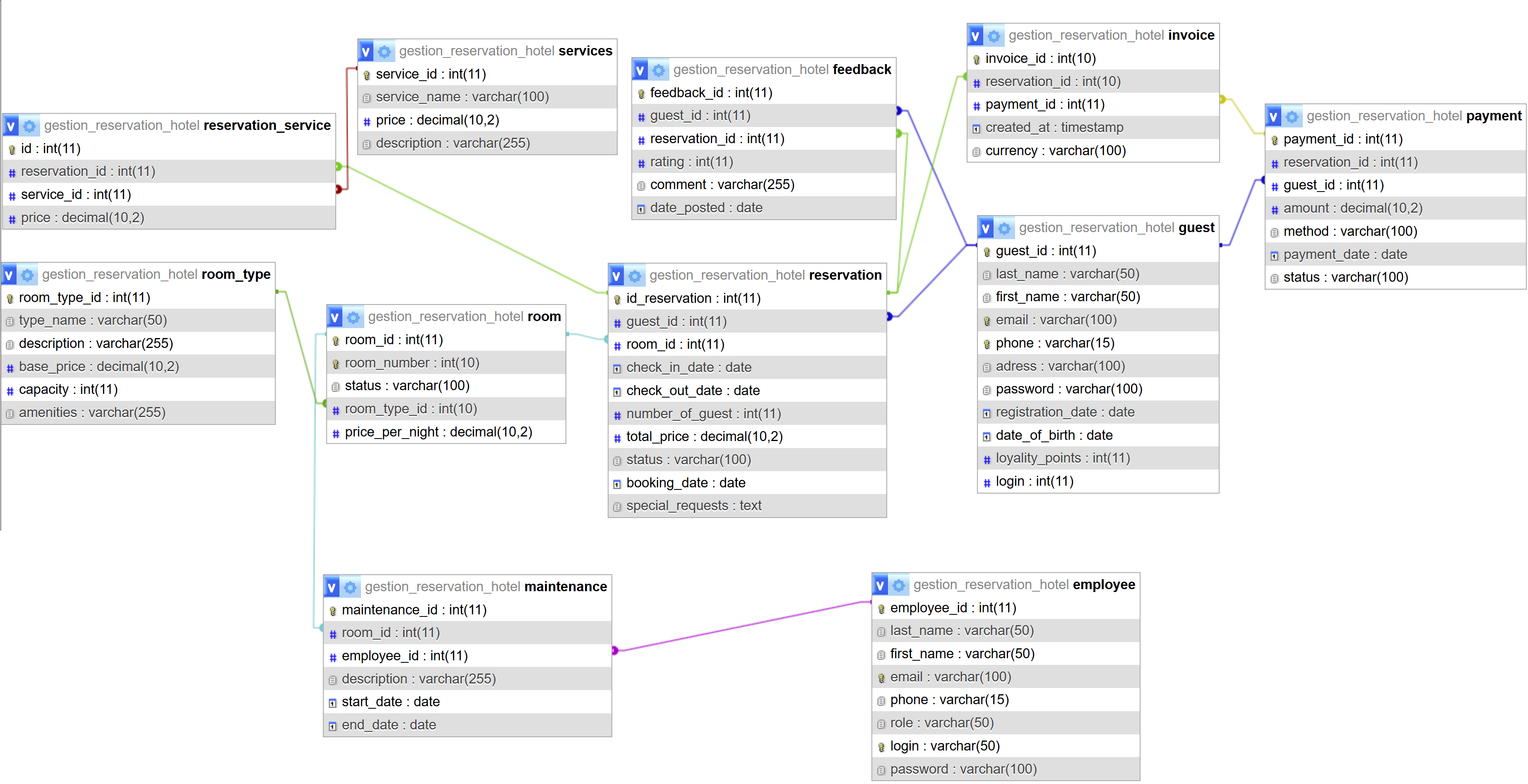
# 4. Database Design

## 4.1 Conceptual Data Model

The conceptual model defines the main entities and relationships between them. The major entities include Guest, Room, Reservation, Payment, Feedback, and Staff.



## 4.2 Relational Model

The relational schema was designed based on the conceptual model, ensuring normalization up to the third normal form (3NF). Each table includes a properly defined primary key and foreign keys to maintain referential integrity.

5. Implementation

The implementation was carried out using HTML, CSS, PHP, and MySQL. The interface was designed for intuitive navigation with a clear menu structure and role-based access control. Each functionality is accessible according to the user’s role.

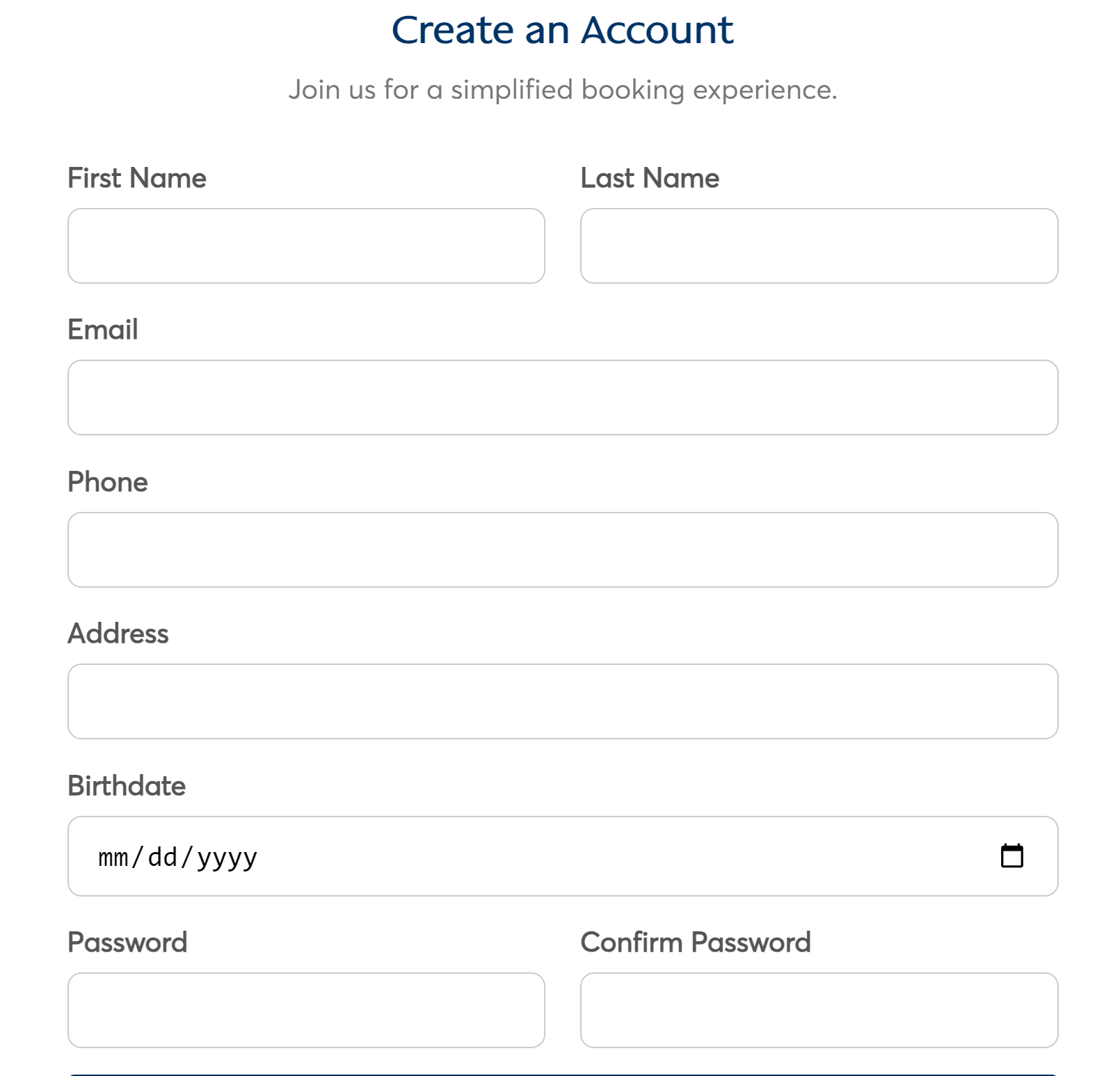
# 6. Implemented Features

## 6.1 Login and User Authentication

Goal:  
make the link between the user who is using the website and the website itself.

How it is done:  
if the user has already registered he only has to put his email and his password to connect to his session.

On the other side, if the user is on his first time login, he must create an account with putting his name, first name, email, address, phone number, a password, to confirm the password and his birth date

Conclusion:  
the login and authentication page makes the link between all the others pages so the program know which column of the database we are refering to

## 

## Register\_process.php :

$password\_hash = password\_hash($password, *PASSWORD\_DEFAULT*);  
  
$stmt = $pdo->prepare("  
 INSERT INTO guest (first\_name, last\_name, email, phone, adress, password, registration\_date, date\_of\_birth, loyality\_points)  
 VALUES (?, ?, ?, ?, ?, ?, CURDATE(), ?, 0)  
");  
  
$result = $stmt->execute([  
 $first\_name, $last\_name, $email, $phone, $address,  
 $password\_hash, $date\_of\_birth  
]);

***Here an exctract of how values are insert into the database.***

***Here, each ‘?’ will be replaced by the correspondent value in the result and the program will also verify if the email given isn’t already used***

## // Tentative de connexion en tant que client $st = $pdo->prepare("SELECT \* FROM guest WHERE email = ?"); $st->execute([$email]); $g = $st->fetch(); if ($g && password\_verify($password, $g['password'])) { // Authentification réussie : client $\_SESSION['user\_id'] = $g['guest\_id']; $\_SESSION['user\_type'] = 'guest'; header('Location: compte\_client.php'); exit; } // Si échec, tentative de connexion en tant qu’employé $st = $pdo->prepare("SELECT \* FROM employee WHERE email = ?"); $st->execute([$email]); $e = $st->fetch(); if ($e && password\_verify($password, $e['password'])) { // Authentification réussie : employé $\_SESSION['user\_id'] = $e['employee\_id']; $\_SESSION['user\_type'] = 'employee'; $\_SESSION['role'] = $e['role']; header('Location: admin\_dashboard.php'); exit; } 6.2 Guest Management

## login\_process.php :

***In this bloc of code, the program verifies if the address email and password entered correspond to a client, an employee or neither and redirects the user in the correspondent page***

Goal:  
each customer should be able to see all their personals information on their profile and modify whatever they want, they must also see their reservations.

Admins could modify whatever they want from clients and as a result delete a client profile

How it is done:

## admin\_gestion\_clients.php:

// ÉTAPE 2 : Construction dynamique de la requête SQL  
$sql = "SELECT \* FROM guest WHERE 1=1";  
$params = [];  
  
// Ajout de la recherche par nom/email/téléphone  
if ($search) {  
 $sql .= " AND (first\_name LIKE ? OR last\_name LIKE ? OR email LIKE ? OR phone LIKE ?)";  
 $searchTerm = "%$search%";  
 $params = array\_merge($params, [$searchTerm, $searchTerm, $searchTerm, $searchTerm]);  
}  
// Ajout du filtre par points de fidélité  
if ($filter\_points === 'high') {  
 $sql .= " AND loyality\_points >= 100";  
} elseif ($filter\_points === 'medium') {  
 $sql .= " AND loyality\_points BETWEEN 50 AND 99";  
} elseif ($filter\_points === 'low') {  
 $sql .= " AND loyality\_points < 50";  
}  
  
$sql .= " ORDER BY guest\_id DESC";

***‘WHERE 1=1’ allows you to easily add conditions with AND***

***LIKE? allows partial search***

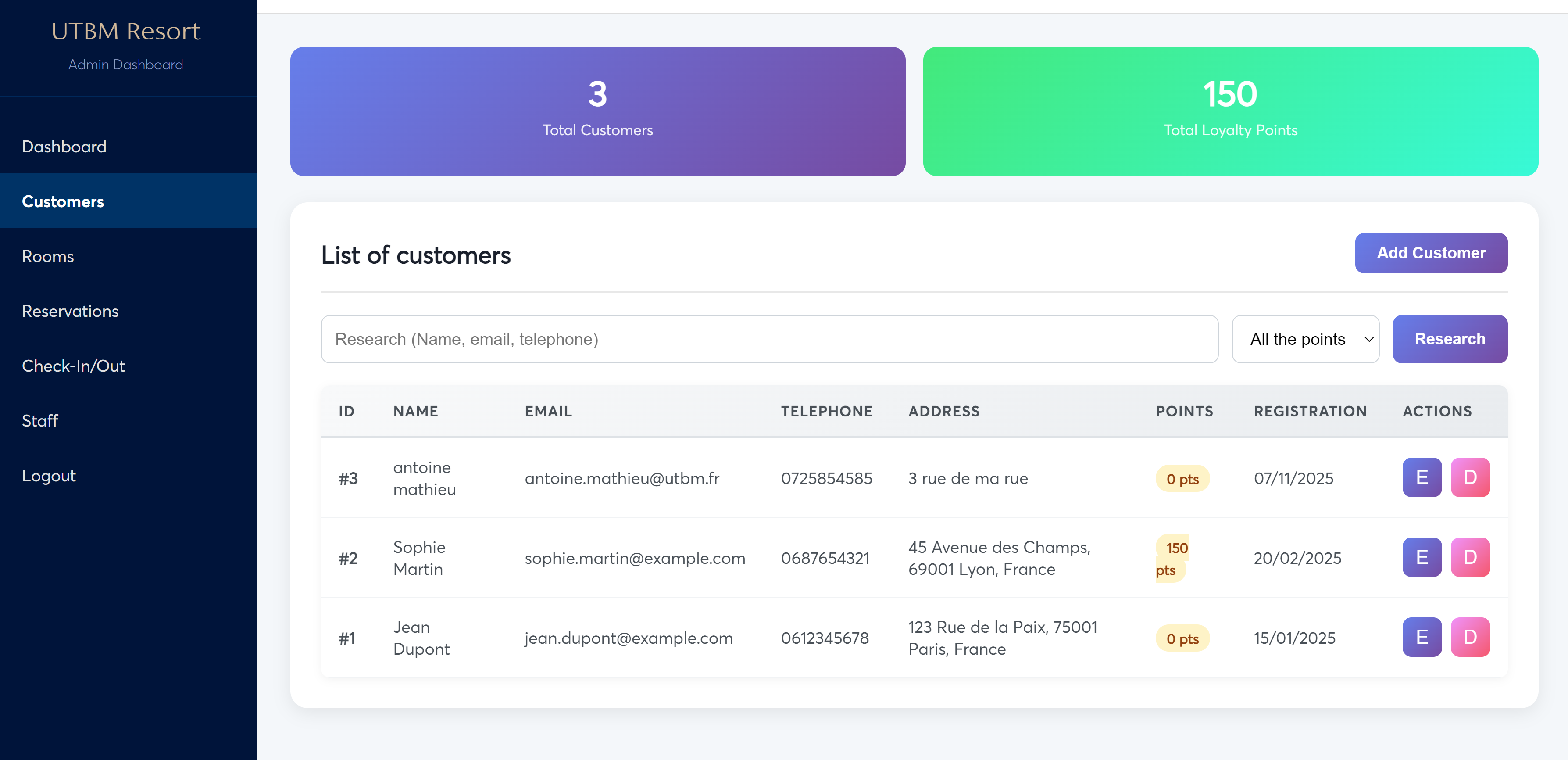
***‘%$search%’: ‘%’ means "any characters before/after"***

***Filter by points:***

***high: VIP customers with ≥100 points***

***medium: regular customers (50-99 points)***

***low: new customers (<50 points)***



## 6.3 Room Management

Goal:  
manage and characterize rooms as easily as possible.

Possibility to add/delete/modify rooms for admins

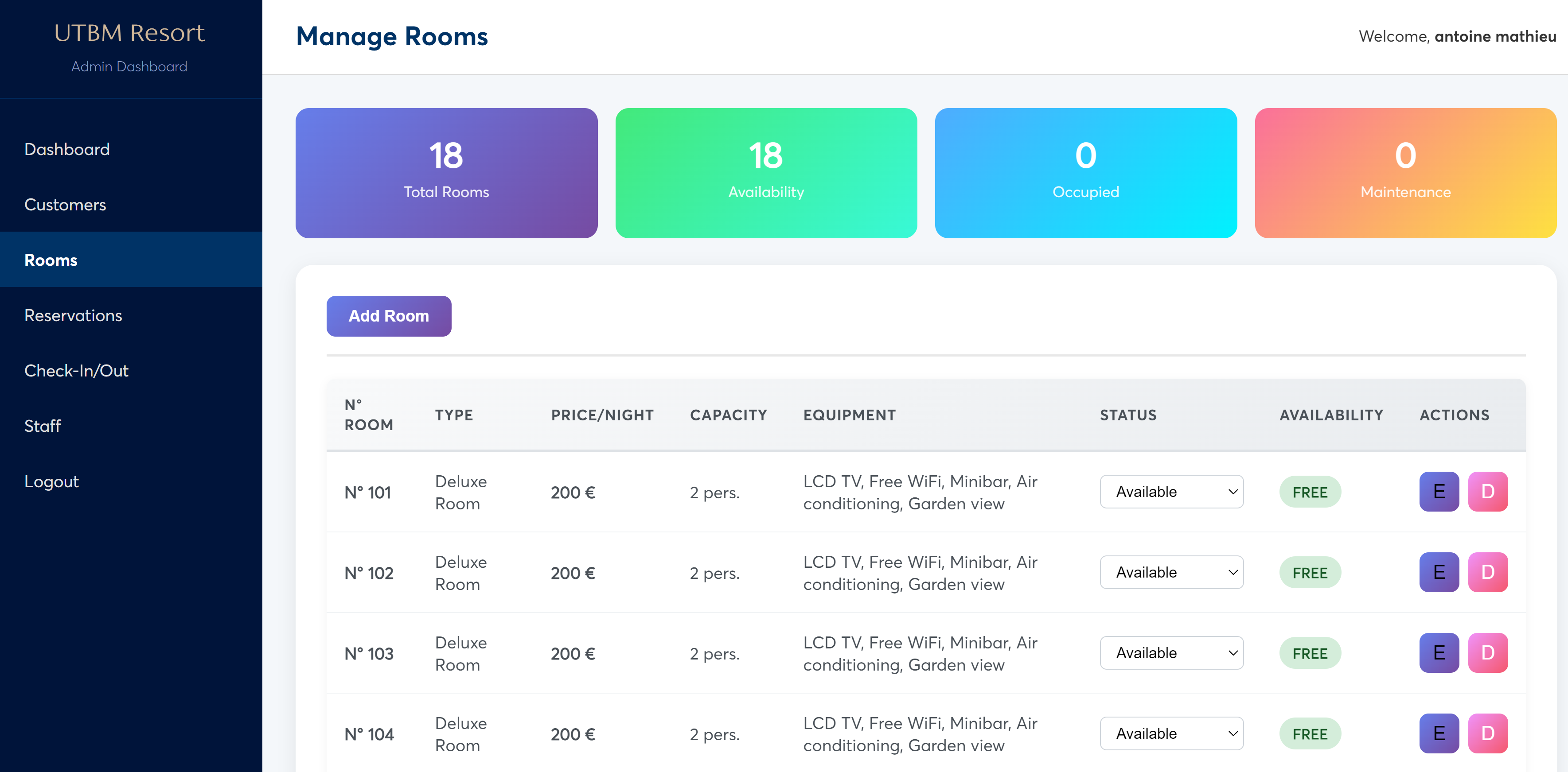
How it is done:

## admin\_gestion\_chambres.php:

case 'add':  
 $stmt = $pdo->prepare("SELECT base\_price FROM room\_type WHERE room\_type\_id = ?");  
 $stmt->execute([$\_POST['room\_type\_id']]);  
 $price = $stmt->fetch()['base\_price'] ?? 0;  
   
 $stmt = $pdo->prepare("INSERT INTO room (room\_number, room\_type\_id, price\_per\_night, status) VALUES (?, ?, ?, 'available')");  
 $stmt->execute([$\_POST['room\_number'], $\_POST['room\_type\_id'], $price]);  
 $\_SESSION['success'] = "Room successfully add.";  
 break;  
  
case 'edit':  
 if (empty($\_POST['price\_per\_night'])) {  
 $stmt = $pdo->prepare("SELECT base\_price FROM room\_type WHERE room\_type\_id = ?");  
 $stmt->execute([$\_POST['room\_type\_id']]);  
 $price = $stmt->fetch()['base\_price'] ?? 0;  
 } else {  
 $price = $\_POST['price\_per\_night'];  
 }

***A switch to distinct if the admin want to add a new room or edit an existent room.***

***In each case a request is created to change the database***



## 6.4 Reservation System

Goal:  
the purpose of the reservation system is to handle everything which permit the user to book a room

How it is done:

## Reservation\_process.php:

if (!empty($validServices)) {  
 $ins = $pdo->prepare("INSERT INTO reservation\_service (reservation\_id, service\_id, price) VALUES (?, ?, ?)");  
 foreach ($validServices as $s) {  
 $ins->execute([$reservation\_id, (int)$s['service\_id'], (float)$s['price']]);  
 }  
}  
  
$pdo->commit();  
  
$\_SESSION['success'] = "Reservation created successfully. (Status: Pending confirmation)";  
header("Location: compte\_client.php?section=reservations");  
exit();

***Different scenario are prepared for each errors that can happen : if the capacity of the room is not enough, if the room selected doesn’t exists, if the check in date and check out date are incorrect. If none error is declared then a SQL query is created to create a new reservation in the table and then redirect the guest in the ‘compte\_client’ page***

## 

## 6.5 Check-in / Check-out Management

Goal:  
handle the check in and the checkout process automatically when the customer arrives and leaves his room

How it is done:

## Process\_checkin.php:

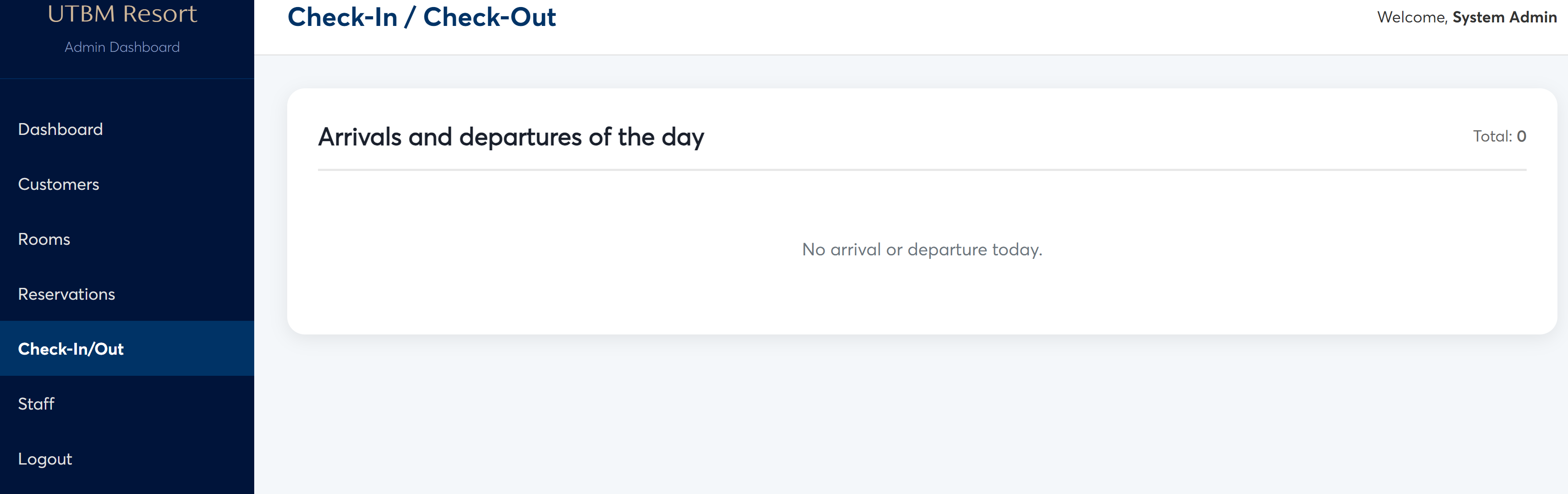
$pdo->beginTransaction();  
  
$stmt = $pdo->prepare("UPDATE reservation SET status = 'checked\_in' WHERE id\_reservation = ?");  
$stmt->execute([$reservation\_id]);  
  
$stmt = $pdo->prepare("  
 UPDATE room r  
 JOIN reservation res ON r.room\_id = res.room\_id  
 SET r.status = 'occupied'  
 WHERE res.id\_reservation = ?  
");

***Modify de state of the reservation to be occupied during the duration the room was reserved***

## Process\_checkout.php:

$pdo->beginTransaction();  
  
$stmt = $pdo->prepare("UPDATE reservation SET status = 'completed' WHERE id\_reservation = ?");  
$stmt->execute([$reservation\_id]);  
  
$stmt = $pdo->prepare("  
 UPDATE room r  
 JOIN reservation res ON r.room\_id = res.room\_id  
 SET r.status = 'cleaning'  
 WHERE res.id\_reservation = ?  
");

***Modify de state of the reservation as completed when customers leave the room and put the status of the room in cleaning with the SQL query ‘ set r.status = “cleaning” ’***



## 6.6 Feedback and Reviews

Goal:  
once your stay is finished, you should be able to leave a review and a mark to the hotel so we can help improving our services

How it is done:

## Submit\_feedback.php:

try {  
 $ins = $pdo->prepare("  
 INSERT INTO feedback (comment, rating, date\_posted, guest\_id, reservation\_id)  
 VALUES (?, ?, NOW(), ?, ?)  
 ");  
 $ins->execute([$comment, $rating, $\_SESSION['guest\_id'], $reservation\_id]);  
  
 $\_SESSION['success'] = "Thank you for your feedback!";  
 header('Location: compte\_client.php?section=reviews');  
 exit();  
  
} catch (Exception $e) {  
 $\_SESSION['errors'] = ["Error when sending the feedback: " . $e->getMessage()];  
 header('Location: compte\_client.php?section=reviews');  
 exit();  
}

***Ask the customer to give a feedback of his stay, this feedback is then insert in the correspondent table. The feedback is only possible if the customer’ reservation is end moreover it is only one feedback per reservation.***

## 

# 7. Results and Evaluation

The system successfully meets the objectives defined in the initial project requirements. All major functionalities were implemented and tested with multiple user roles. The application provides a smooth and efficient reservation experience for both guests and hotel staff.

A perspective to upgrade the website would be the payment method when the customer reserve a room and the stats analysis linked to the feedback so statistics could be created thanks to a lot of comments and rates

# 8. Conclusion

This project allowed us to design and develop a hotel reservation system, integrating both user functional requirements and the technical constraints related to data management. Through requirement analysis, system modeling, and the implementation of various features, we were able to deliver a solution that is intuitive, secure, and scalable.

The project also highlighted the importance of data structuring and careful planning in software development, as well as the need to adapt tools and technologies to the specific needs of the project. Looking ahead, the system could be enhanced with advanced functionalities such as real-time availability management, automated notifications, and a statistical analysis module to optimize hotel operations.

Overall, this project provides a solid foundation for future developments and demonstrates the practical application of theoretical concepts studied during our training.