People's Democratic Republic of Algeria Ministry of Higher Education and Scientific Research
M'hamed Bougara - Boumerdès University

Faculty of Sciences
Computer Science department

Domain : Computer Mathematics
Program : Computer Science
Specialty : Computer System
N ° of the specialty authorization decree: Decree No. 872 of 07/26/2016

Final graduation memory in view of obtaining the academic license diploma

---

Theme

Design and Development of a Hotel Management Application

Carried out by:

- BENNACEUR Amel
- BENNOUI Marya Halim
- KHEDIMI Hadil

Defended on: /06/2026 Before the jury composed of:

- Mr. Ali Chaouche : Examiner
- Dr. Djeddai Selma :Supervisor
- Mr. Kelzou Omar : External Supervisor

Academic Year: 2025/2026

Acknowledgments

After Bismillah, First we thank Allah for educating us and guiding us, giving us the energy to complete our project.
We would like to express our deepest gratitude to our supervisor Dr.S.Djeddai, for her patience, guidance, and everything she has done for us throughout this project. Her support and valuable advice have been essential in bringing this work to completion.

We also extend our sincere thanks to the examiner, Mr. Chaouche, for the honor of evaluating our work.

We are grateful to everyone who helped us complete our project until the end, whether through direct assistance or moral support.

Finally, we thank our families and friends for their unconditional support and constant encouragement. Their belief in us has been a source of strength throughout this journey.

Dedication

To our mothers and fathers, who never stopped believing in us even when we doubted ourselves.
And to ourselves, Amel, Hadil, Marya, for every second of effort, every tear, and every small victory along the way.

Abstract

Web applications are widely used today in many domains to meet our needs and requirements, making things effective and easy-to-use. Web apps make communication with clients easier and offer many features online. This project aims to develop a web application that allows users to navigate, search, and reserve rooms online rather than going to the hotel. Not forgetting receptionists who can handle received reservations and manage the state of rooms. A manager who has access to all features and can add or remove a room. We will use UML to design our diagrams (use case, class, and sequence). To complete our app and make it ready to use, we used HTML for structure, CSS for design, JavaScript to make interactions dynamic, PHP for server side, and MySQL for database**.**

Keywords: Hotel, Room reservation, web application.

Content

List of figures VI

List of tables VII

General Introduction 1

Chapter 1 study of the existence 2

Introduction 2

1.1 Presentation of host organization 2

1.2 State of art 3

1.3 Job role study 4

1.4 Documentary Study 4

1.5 Problem statement 6

1.6 Solution 6

Conclusion 6

Chapter 2 Conception and Specification 7

Introduction 7

2.1 Requirement definition 7

2.2 Use case diagram 9

2.3 Sequence diagrams 11

2.4 Class diagram 16

2.5 Transition to Relational Model 17

2.6 Relational Data Model 18

Conclusion 18

Chapter 3 Implementation 18

Introduction 18

3.1 Development Environment and Programming Languages 19

3.2 Application interfaces 20

Conclusion 23

Bibliographie 26

List of figures

2.1 Use Case Diagram of Hotel Management System 10

2.2 Authentication sequence diagram 12

2.3 Sequence diagram of reservation process 13

2.4 Sequence diagram for room additions. 14

2.5 Sequence diagram for check-in. 15

2.6 Class diagram of Hotel Management System 16

3.1 Admin's interface 21

3.2 Manager interface, room management page 22

3.3 Receptionist's interface for all reservations 22

3.4 Interface client - list of online reservations 23

3.5 Home Page, Reservation Online (Interface Client) 23

List of tables

1.1 Hotel Documents 5
3.1 Programming Languages and Tools Used 20

General Introduction

In an era defined by digital transformation and instant connectivity, the hospitality industry faces increasing pressure to modernize its operations. Traditional methods of managing hotel reservations, which depend on the use of office software such as Excel and Word, along with telephone calls, are no longer sufficient to meet guest expectations for speed, precision, and convenience. Today's travelers demand seamless 24/7 access to room availability, real-time booking confirmations, and personalized experiences, all from the palm of their hand. For hoteliers, this transition necessitates moving away from disparate, non-integrated tools toward unified, reliable, and efficient web application.

This report documents the work carried out during our final year project, the primary objective of which was to design and implement a complete hotel reservation system. The goal was not only to create a booking interface, but to develop a full-featured platform that unifies the entire guest journey from room search and online reservation to check-in, payment processing, and administrative management. The system addresses key challenges such as preventing double bookings, managing room inventory in real time, automating billing, and providing hotel staff with a centralized dashboard to oversee all reservations and customer data.

Regarding the structure of this report, it is organized into three chapters. The first chapter presents the Host Organization and covers the study of the existence. The second chapter focuses on the Specification and Design. The third and final chapter discusses the Implementation, including the technologies used and the testing results.

Chapter 1 Study of the existence

Introduction

This chapter presents the Preliminary Study conducted to understand the existing environment before proposing our new system. It covers the host organization presentation, state of art, job role study, documentary study, problem identification, and the proposed solution.

1.1 Presentation of host organization

Our practical work was carried out at WebShore Algeria, a web agency founded in 2016 by Mr. KELZOU OMAR and located in Boumerdes, Algeria. The agency specializes in creating custom digital solutions including websites, mobile applications, web dashboards, reservation systems, management software for sale and inventory, digital marketplaces, and training platforms. To ensure quality results, WebShore follows a structured methodology that includes needs analysis, design, development, testing, deployment, training, and maintenance. During our internship, we worked on the design and development of a hotel management web application called “HEAVEN”.

1.2 State of the art

In the hotel industry, there are many solutions for hotel management and reservation management. These solutions include various software tools and online platforms, which fall into two main categories:

1. Online Travel Agencies (OTAs):
   They are online platforms that allow guests to search for and book hotel rooms from outside the hotel. These platforms send booking information directly to the hotel.
   For example:
   - Booking.com: A global online travel agency that allows guests to search, compare, and book hotels based on price, services, and reviews. Reservations are sent directly to the hotel, and payment can be made online or upon arrival.[1]
   - Expedia: A major American online travel agency that owns multiple brands such as Hotels.com [16] and Orbitz. Hotels must sign a contract with Expedia to appear across all its platforms.[13]
   - Airbnb: Originally a platform for renting apartments and homes, now also includes hotel listings. Guests must pay through the platform after choosing their accommodation.[15]
2. Property Management Systems (PMS):
   They are software used inside the hotel by staff such as receptionists and managers. It helps with tasks like checking guests in and out, managing reservations, and generating bills and reports. For example:
   - Oracle Opera: A property management system designed for luxury hotels and large chains, available both in the cloud and on-premise. It offers strict access controls, comprehensive reporting, and tools for check-ins, reservations, and billing.[17]
   - Mews: A cloud-based property management system accessible via web browsers and mobile devices. It is known for being flexible and user-friendly, with features like contactless check-in and an open API for integrations.[14]
   - Cloudbeds: A unified cloud platform that combines property management, channel management, and a booking engine. It is popular among independent hotels and hostels for its all-in-one approach and ease of use.[18]

1.3 Job role study

Hotel management requires a diverse workforce, with each member fulfilling specific duties. The manager oversees financial performance, budgeting, recruitment, and staff coordination. Receptionists, who work in shifts, manage guest check-ins, check-outs, payments, room assignments, and requests, while also submitting police records for foreign guests and noting preferences. Housekeeping staff clean rooms, inspect furniture and equipment, report room status, and handle laundry services for paying guests. The kitchen manager oversees food services, menu planning, cooking supervision, hygiene control, and ingredient quality and cost management. Security personnel protect the premises, intervene in conflicts, and monitor suspicious activity. Additional staff may be present depending on the hotel's specific strategy.

1.4 Documentary Study

The documentary study is a preliminary investigation that allows us to identify which documents the hotel used in the past, or still uses to this day. The table below presents some of the documents that many hotels continue to use until now.

Document Title	Used By	Origin	Destination	Purpose
Police Registration Form	Reception	Guest	Police	Required for foreign guests; submitted daily to Police
Welcome Card	Reception	Reception	Guest	Guest identity and hotel services information
Registration Card	Reception	Guest	Reception	Captures guest name, ID, address for legal/billing
Family Book	Reception	Guest	Reception	Required for families with children to verify relationships
Customer Obligations	Reception + Security	Management	Guests	Required ID, payment at check-in, minimum age
Damage Penalty Policy	Reception +House-keeping	Management	Guests	4 levels of damage penalties
Check-out Policy	Reception	Management	Guests	Check-out by 12:00 or pay extra night
Laundry Order Form	Reception +House-keeping	Guest	Reception	Guest lists items for cleaning
Document Title	Used By	Origin	Destination	Purpose
Room Status Re-Port	House keepig +Recep-tion	Reception	Housekeeping	Shows rooms: Occupied, Vacant, Dirty, Clean
Tour Schedule	Reception + Concierge	Tour Oper-ator	Reception	Records tour arrivals(Part, Company, Date, Duration)

Table 1.1: Hotel Documents

1.5 Problem statement

Based on the state of the art, it is evident that the hospitality industry offers numerous solutions for hotel and reservation management. However, these solutions are generic and are not tailored to the specific needs of individual hotels. Consequently, a hotel wishing to adopt such a system is likely to encounter either an excess of unnecessary functionalities or a lack of essential features, precisely because these solutions are designed to be general rather than specific. Therefore, the central research question of this study is: how can we design and develop a web application for hotel reservation management that effectively addresses the specific requirements of a particular hotel?

1.6 Solution

To address the identified problem, this study proposes a secure, integrated web-based information system that centralizes all hotel reservations data.
The solution consists of four main components:

1. An online reservation module for clients.
2. A centralized database for all reservation and client data.
3. A role-based internal interface for staff.
4. A real-time synchronization mechanism for updates.

This system aims to eliminate redundant data entry, accelerate financial reconciliation, improve the client booking experience, and support better management decisions.

Conclusion

To conclude, this preliminary study has revealed that although many solutions are available for reservation management, they are not specific to each hotel's unique needs. From this, we conclude that there is a clear need for a more efficient, reliable, and hotel-specific system to replace existing generic solutions. In the next chapter, we will present the conception and specification of our proposed system.

Chapter 2
Conception and Specification

Introduction

In this chapter, we will present the process of conception and modeling of our web application. First, we will present the used method UML then use it in different diagrams (use case, class, sequence). So this chapter will help us presenting functionalities of our web application and showing how it is used.

2.1 Requirement definition

Requirement definition is a primary step in the completion of a project. It helps us define the functionalities that the application will provide to the user and to better specify them, particularly in relation to the initial ideas of what the site will offer. This step is crucial for creating the use case diagram, where each functional requirement translates into a specific case for an actor.

2.1.1 Actor identification

- **Visitor**: The visitor can consult the list of available rooms and create an account to become a registered client.
- **Guest**: The guest can consult rooms, create a reservation, modify an existing reservation, or delete a reservation.
- **Receptionist**: The receptionist manages all reservations, perform check-in and check-out operations, and validates online reservations made by client.
- **Director:** The director manages rooms (add, modify, delete) and manages contracts.
- **Accountant**: The accountant consults reservation information and details for financial tracking purposes.
- **Administrator:** The administrator is the main actor of the application. They have access to all functionalities of the other actors, as well as additional management functions including accounts management and application management.

2.1.2 Functional requirements

This is a description of the application's features and functionalities, so it must allow for:

- To the Visitor:
  - Consult rooms.
  - Create account.
- To the guest :
  - Consult rooms
  - Create a reservation
  - Modify reservation
  - Delete a reservation
- To the receptionist:
  - Reservations management.
  - Check-in.
  - Check-out.
  - Validation of online reservation.
- To the director:
  - Room management.
  - Contract management.
- To the accountant:
  - Consultation of reservations information and details.
- To the admin:
  - Accounts management.
  - Application management.

2.1.3 Non-functional requirements

These correspond to the expected performances that the application must ensure to facilitate and improve the user experience:

- **Performance**: the system must be optimized and respond quickly to the user
- **Reliability**: consistent operation without errors.
- **Relevance**: response corresponds to the action performed and selected.
- **Usability**: the system must be comfortable and simple to use.

2.2 Use case diagram

A use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. Figure 2.1 describes the use case diagram of our system.

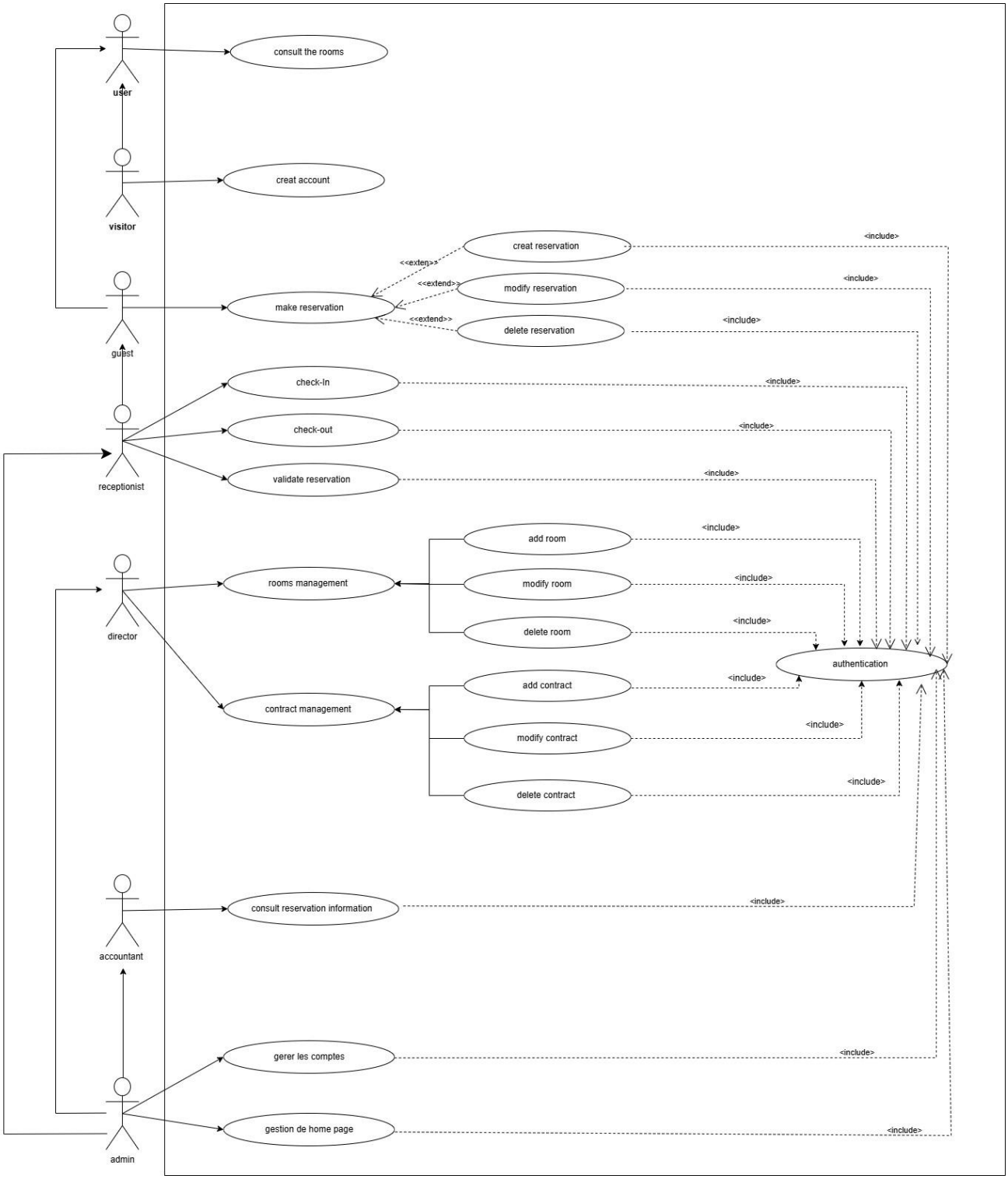


Figure 2.1: Use Case Diagram of Hotel Management System

2.3 Sequence diagrams

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them in the order in which they occur. This allows for the graphical specification of simple runtime scenarios. A system sequence diagram should specify and show external actors, the messages invoked by these actors, any return values associated with previous messages, and any loops or iteration areas. Here, we present the four most important sequence diagrams, as page limitations prevent us from showing all of them.

2.3.1 Authentication sequence diagram

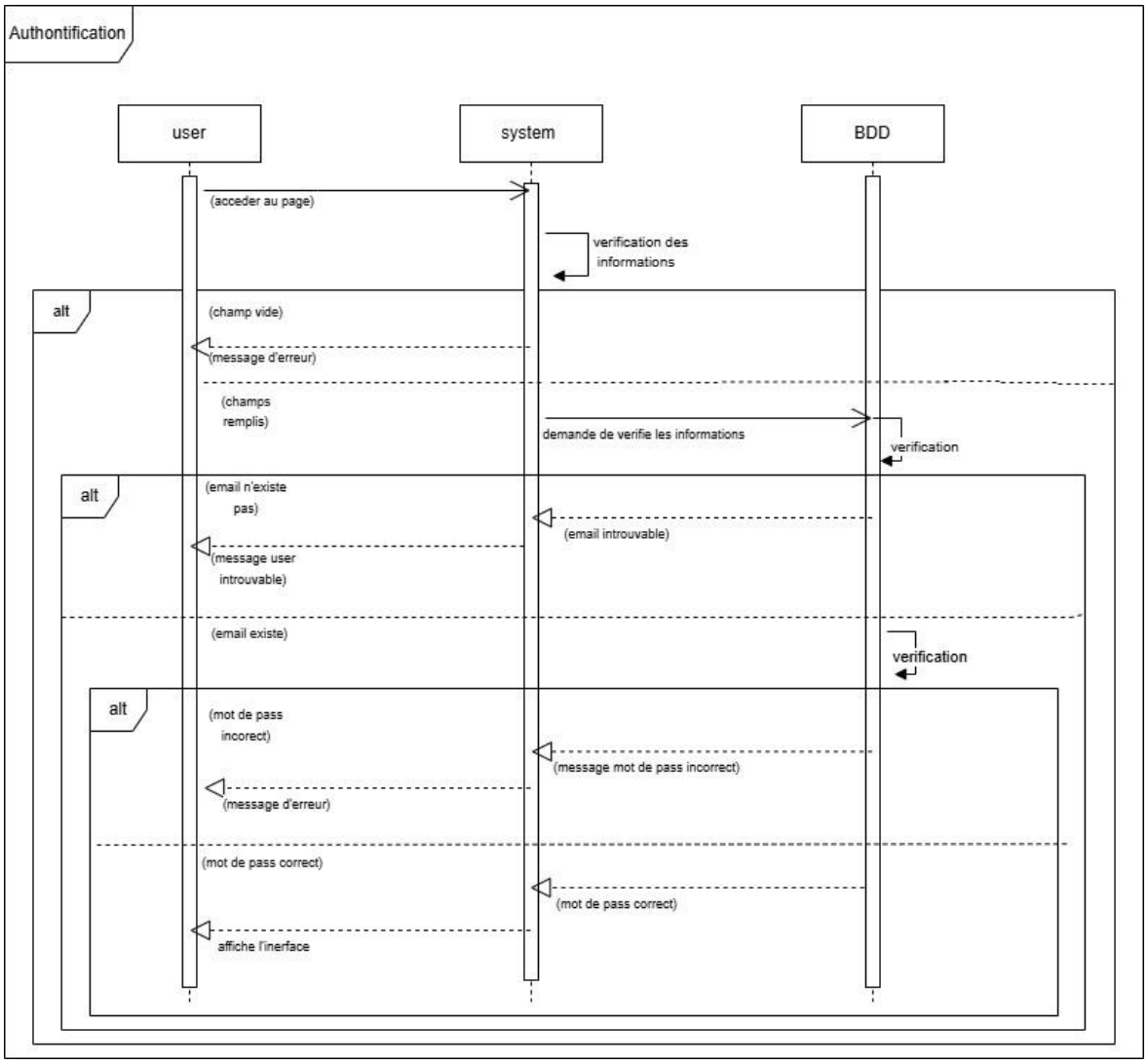


Figure 2.2: Authentication sequence diagram

This sequence diagram shows how a user gets authenticated to access reservation features. The system checks if the user entered the information and if all fields are filled out. It also checks if the email and password are valid. If there is a problem it shows an error message, if not, the user can access the system interface. The system verifies entered information. The user and database are also involved in this authentication process. The system checks all fields. The email and password are validated. If an error occurs an error message is displayed to the user. Otherwise the user successfully accesses the system interface. The user accesses reservation functionalities after authentication.

2.3.2 Creation of a reservation sequence diagram:

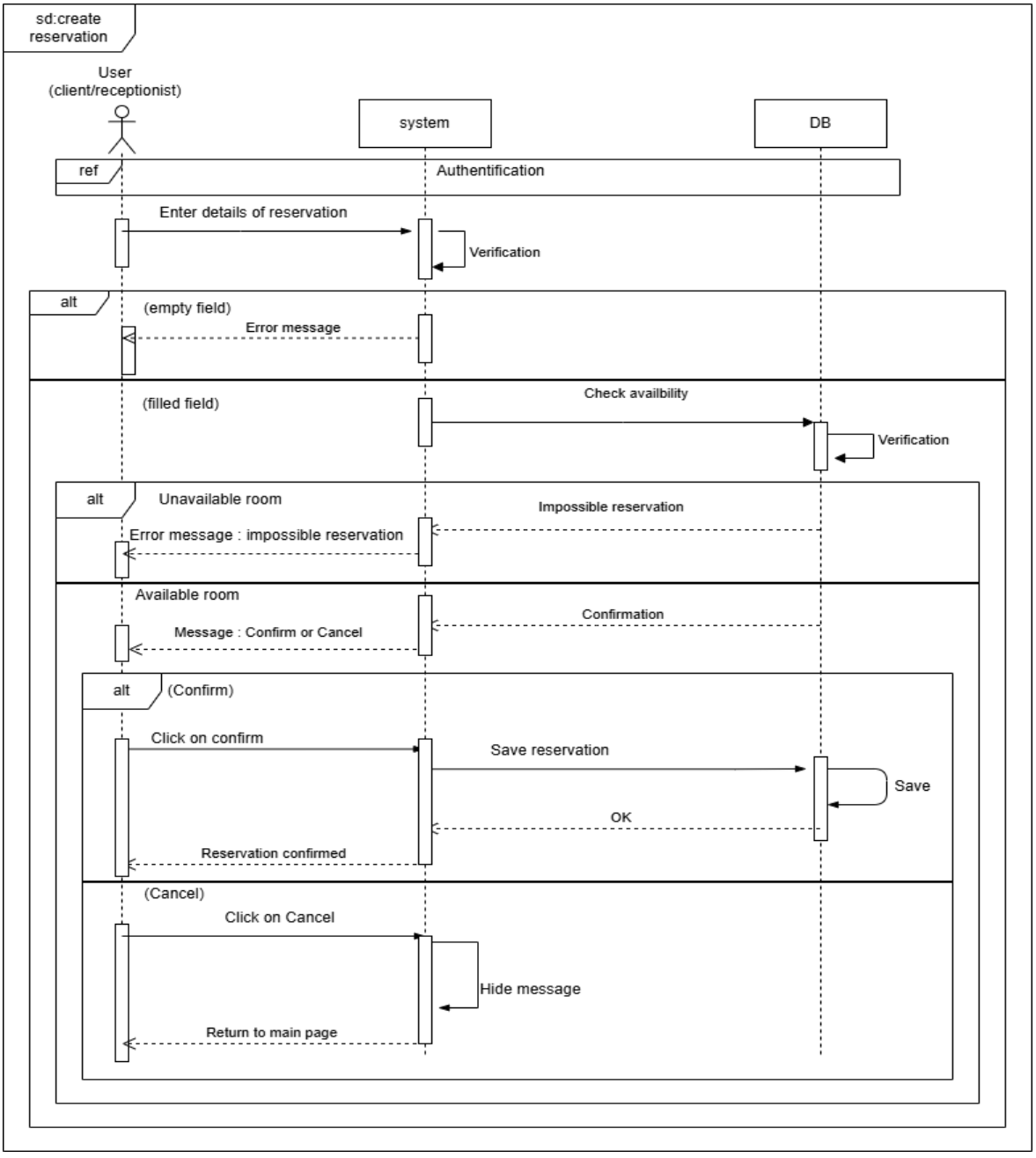


Figure 2.3: Sequence diagram of reservation process

Here we have a sequence diagram of the reservation process, where everything goes between the user whether it is a client or a receptionist and the system with database. So, user enters reservation details that system will verify their availability;

If a field is empty, it will show an error message, otherwise check availability of chosen room and its state. But everything happens after authentication.

2.3.3 Add room sequence diagram:

This diagram represents the process of adding a room, which is similar to the process of adding a contract and adding a promotion

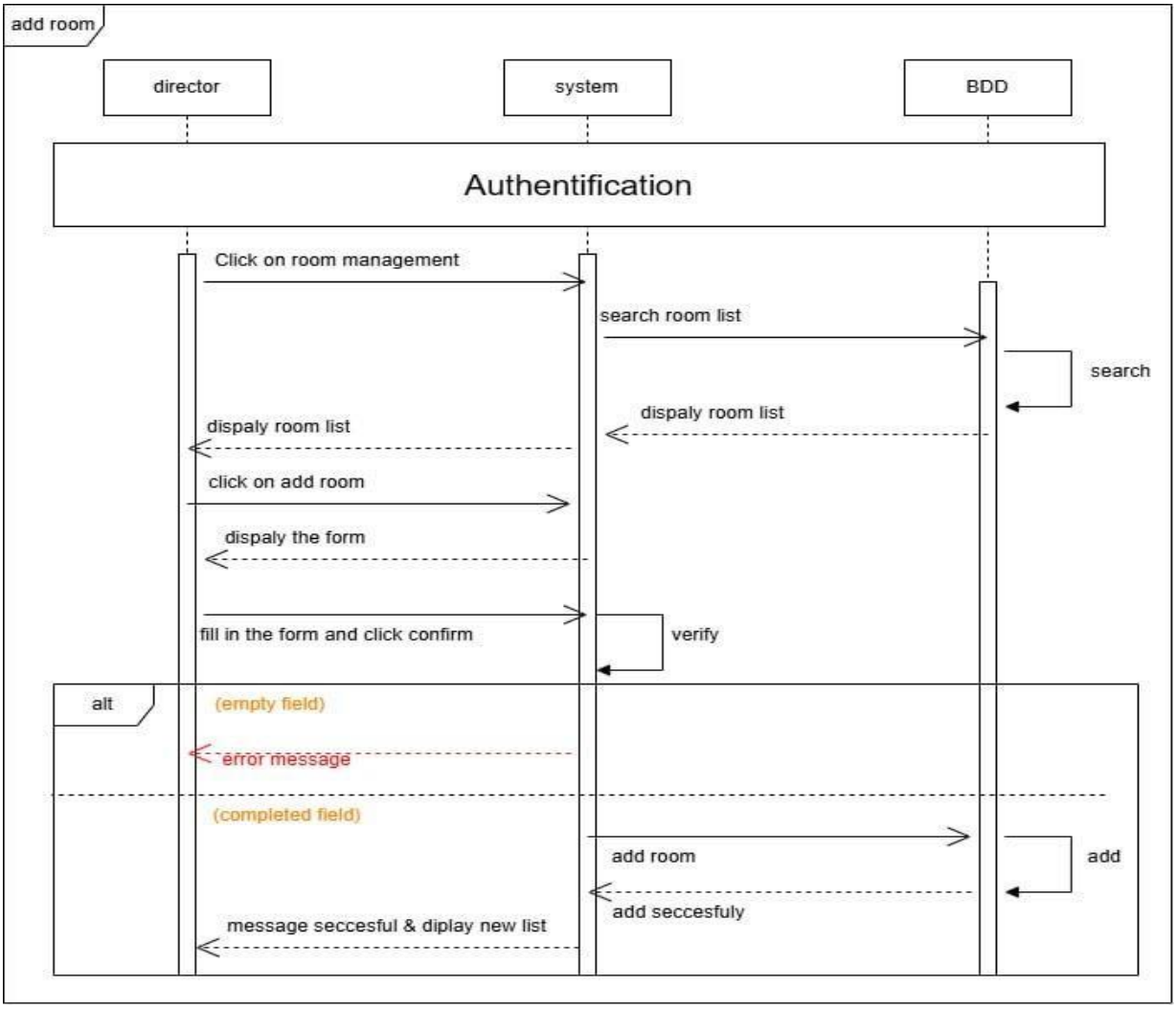


Figure 2.4: Sequence diagram for room addition

2.3.4 Check-In sequence diagram:

The following diagram represents the check-in process, which is similar to the check-out process.

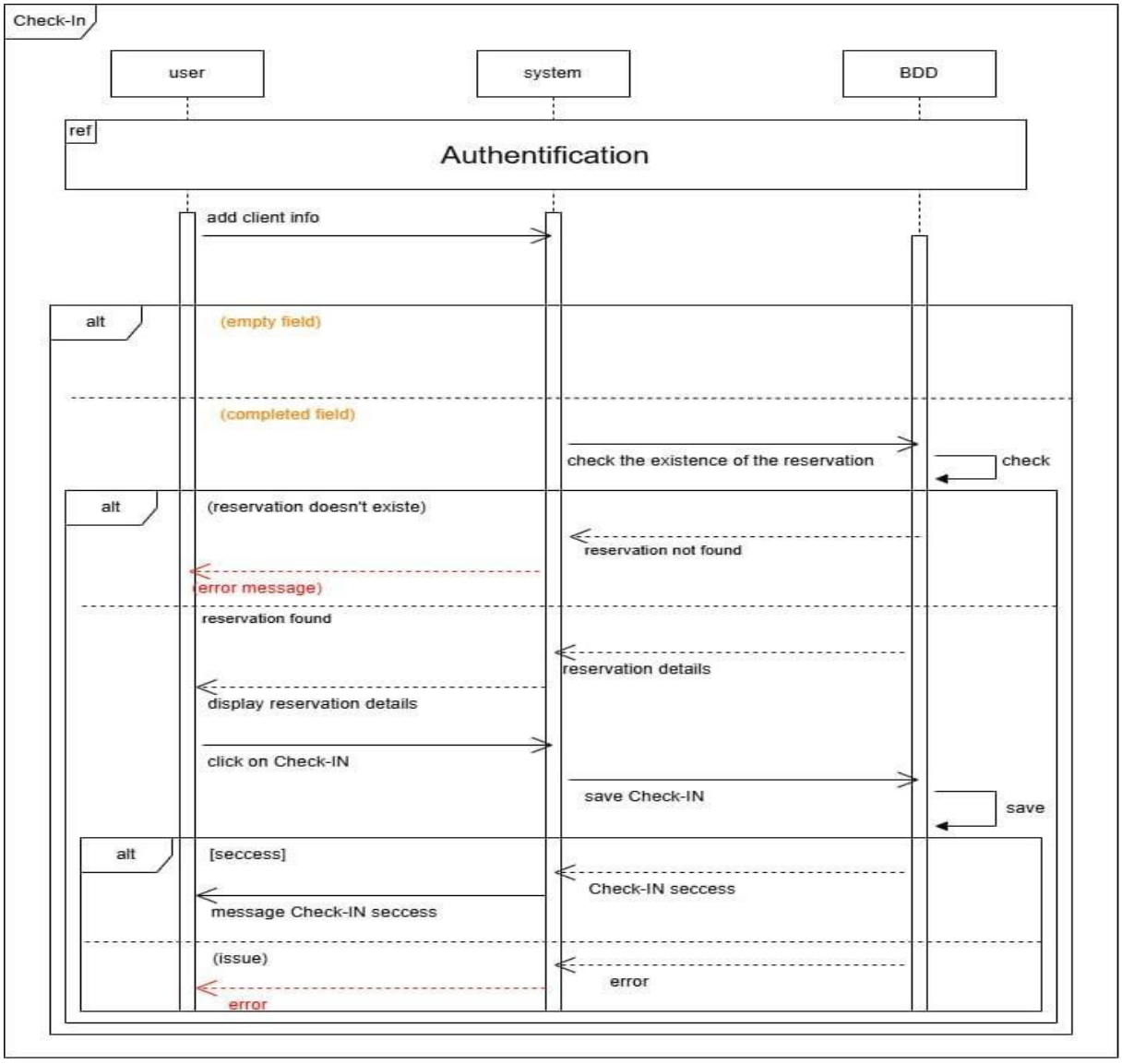


Figure 2.5: Sequence diagram for check-in

2.4 Class diagram

A class diagram shows a set of classes, interfaces, and collaborations as well as the associations among them. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system.

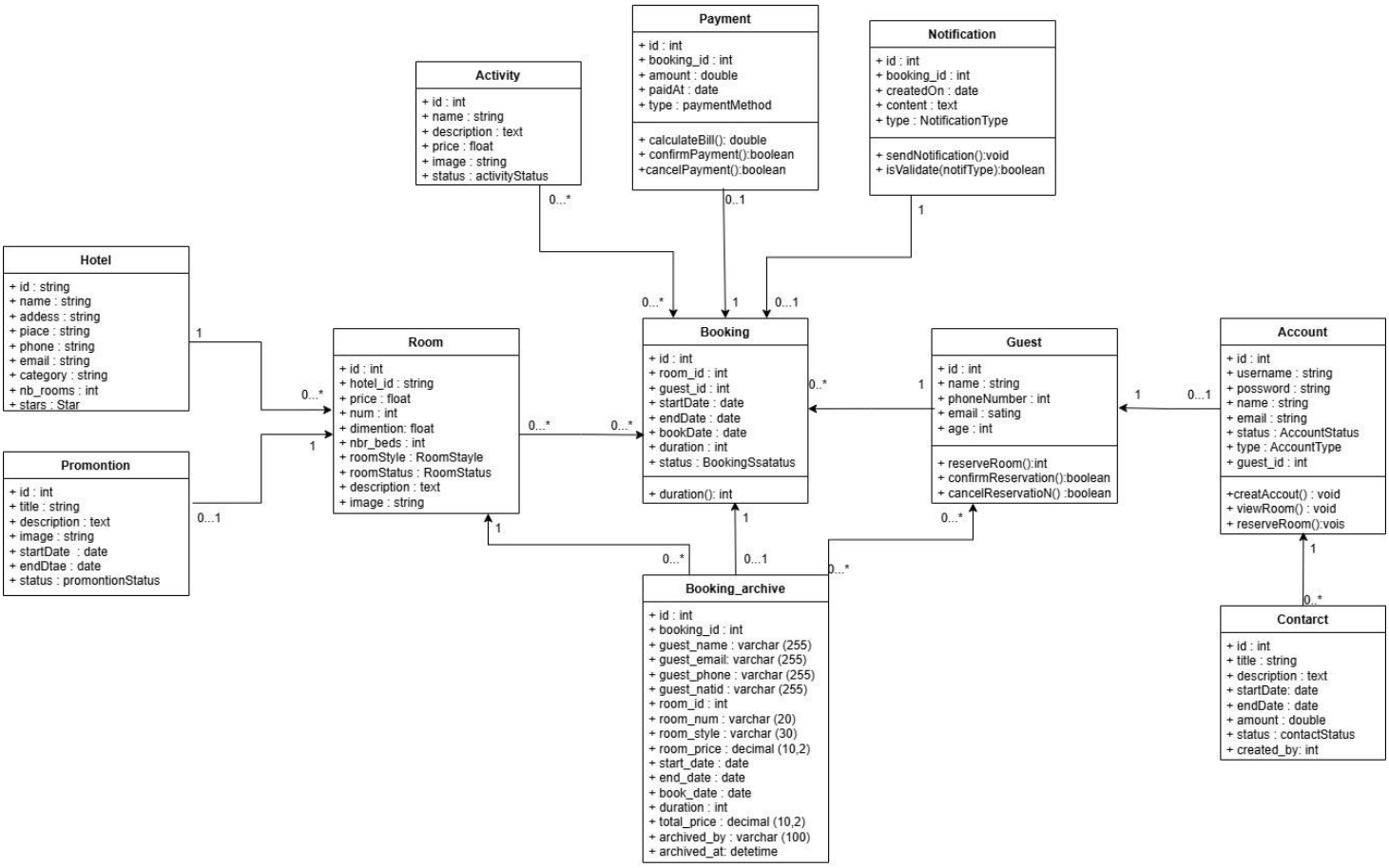


Figure 2.6: Class diagram of Hotel Management System

2.5 Transition to Relational Model

Rule 1: Translation of Classes

- A class becomes a table
- Class properties become table attributes
- The identifier becomes a primary key

Rule 2: Translation of Inheritance Relationship

- Create a table for each subtype
- Each table migrates the key from the parent class to the child classes
- Add generic and specific attributes

Rule 3: Translation of a 1-N Association

- Include the key as a foreign key in the relation whose maximum multiplicity is (n)
- Use the primary key of the other relation

Rule 4: Translation of an M-N Association

- Create a new relation
- The primary keys of the participating relations become foreign keys of the new relation
- Association class attributes are inserted into this new relation if necessary

2.6 Relational Data Model

- Hotel (id_hotel, name, address, place, phone, email, category, nb_rooms, stars)
- Room (id_room, #hotel_id, price, num, dimension, nbr_beds, roomStyle, room-Status, description, image)
- Guest (id_guest, name, phoneNumber, email, age)
- Booking (id_booking, #room_id, #guest_id, startDate, endDate, bookDate, duration, status)
- Payment (id_payment, #booking_id, amount, paidAt, type)
- Notification (id_notification, #booking_id, createdOn, content, type)
- Contract (id_contract, title, description, startDate, endDate, amount, status, created_by)
- Activity (id_activity, name, description, price, image, status)
- Promotion (id_promotion, title, description, image, startDate, endDate, status)
- Booking_archive (id_archive, #booking_id, guest_name, guest_phone, #guest_id, #room_id, room_num, room_style, room_price, startDate, endDate, bookDate, duration, total_price, archived_by, archives_at)

Conclusion

From this chapter, we have completed the conception of our system using UMLbased modeling. In the next chapter, we will implement the system based on the diagram and specifications defined here.

Chapter 3 Implementation

Introduction

In this chapter, we will dive in how really our web application was created and designed as it is a bridge between conception and development. First, we will show software and programming languages we used in releasing our project, giving a brief presentation of each .Then, we will give some interfaces of our application that was made to facilitate use for our users and provide important features.

3.1 Development Environment and Programming Languages

The development of our application was carried out using several well-known programming languages and software tools, which are listed in the table below.

Name & Logo	Description
HTML5	Used to structure the content and layout of web pages. [3]
CSS3	Used for styling the user interface (colors, fonts, responsive design).[4]
JavaScript	Used to add interactivity (form validation, dynamic content).[5]
PHP	Used for backend logic, authentication, and database connection.[6]
MySQL	Used as the database to store rooms, reservations, guests, and users. [7]
WAMP	Used as a local server environment to run PHP and MySQL on Windows.[8]
VS Code	Used as the code editor for writing and debugging the application. [9]
Git	Used for version control to track changes and collaborate. [10]

Table 3.1: Programming Languages and Tools Used

3.2 Application interfaces

In this section, we present the main interfaces of our application, “Heaven”. These screens illustrate the distinct views and functionalities available to each type of user, specifically the admin, receptionist, director, and client. Each interface is designed to provide an intuitive experience tailored to the user’s role, from room research and reservation for client, to reservation validation for receptionist, and overall management for director and administrators.

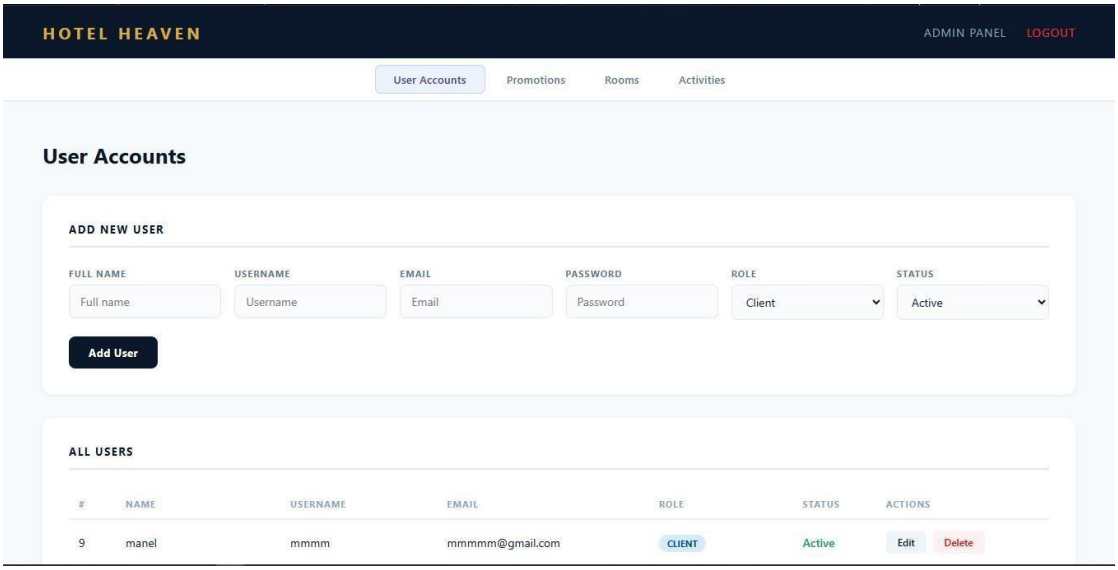


Figure 3.1: Admin’s interface

Promotion interface: First, we have where to add a promotion, providing title, description, start and end dates, status, and image. After confirming it, it’ll be added directly under it in a promotion list.

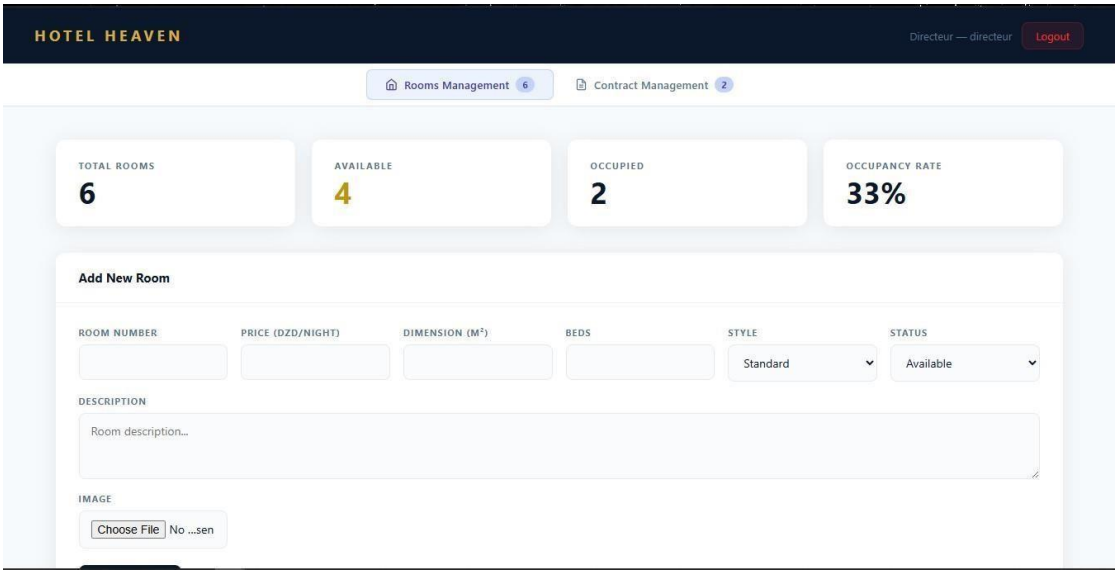


Figure 3.2: Manager interface, room management page

From this interface, manager can check total rooms number, availability and occupation and occupancy rate. Then whenever needed, he can add or delete a room; providing: number, price, dimension, number of beds, style, status, description and image.

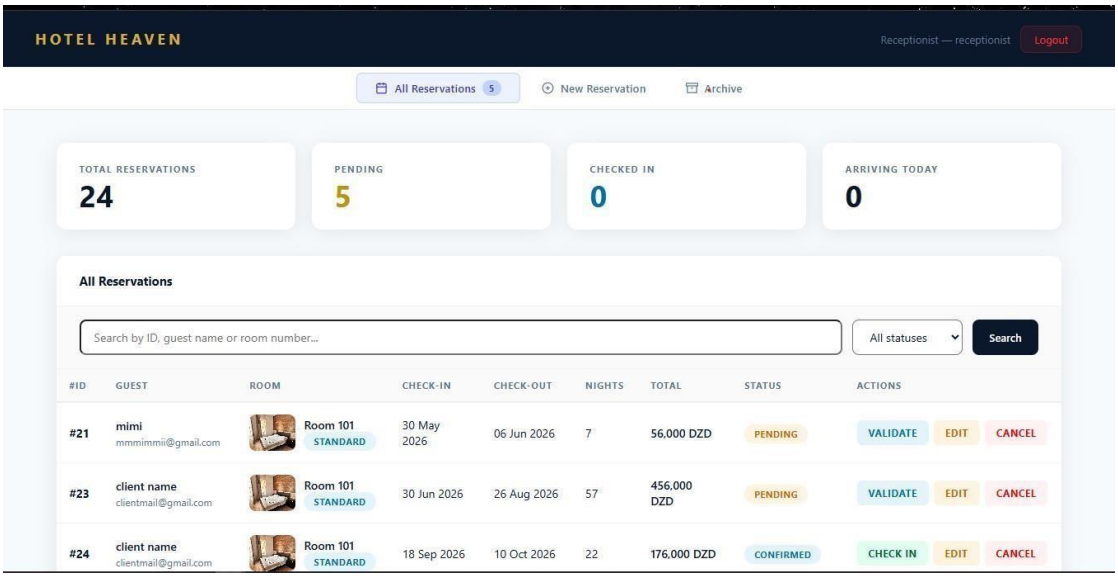


Figure 3.3: Receptionist's interface for all reservations

From this interface and for each day, the receptionist can check the total of reservations, pending, checked-in, and those arriving today. Then, in case a guest comes, the receptionist can verify their reservation by entering their information, to validate, edit or cancel.

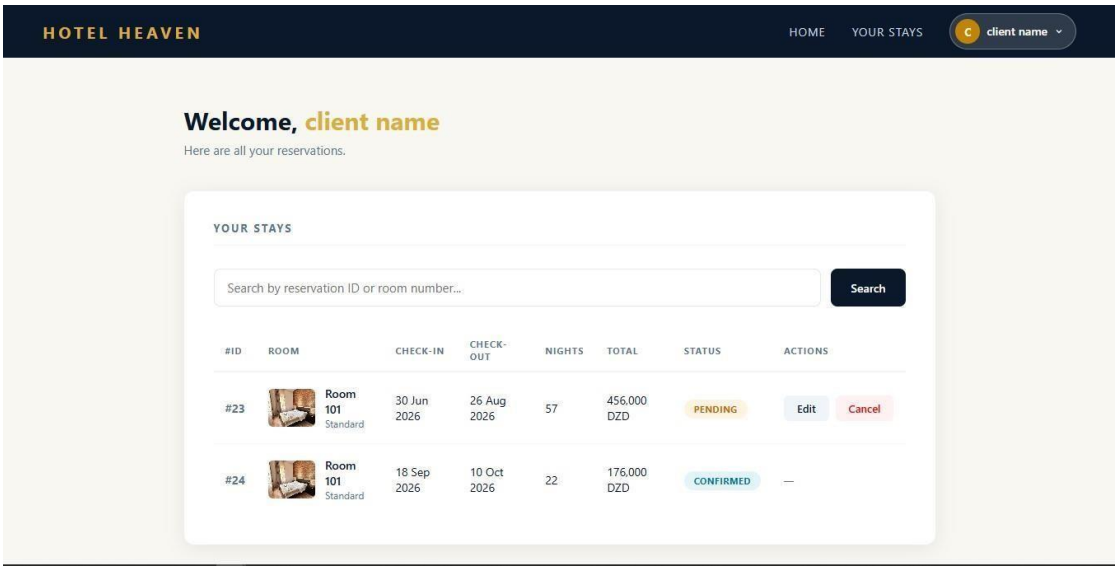


Figure 3.4: Interface guest - list of online reservations

In this interface, called the “guest interface”, the guest can check all his reservations, with all the necessary details, such as room, dates of reservation, prices, etc., although it can be edited or canceled.

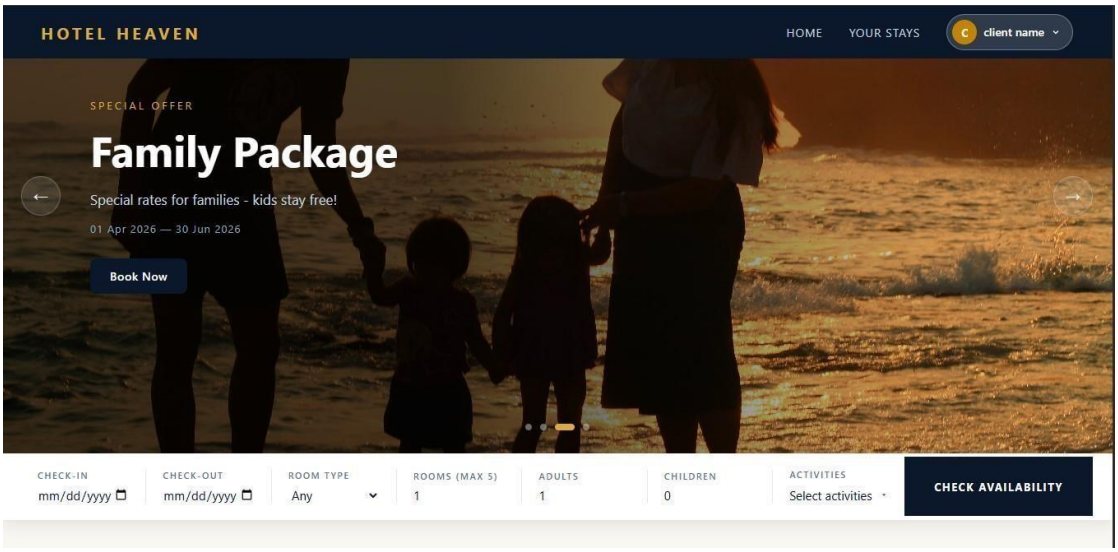


Figure 3.5: Home Page, Reservation Online (Interface guest)

Here, the guest can make reservations by entering his information, and check the availability of the room he chose. After confirming it, it will be displayed automatically in the receptionist's interface so that he confirms or rejects it.

Conclusion

The implementation of a hotel management system demands a methodical approach and the integration of modern technologies to produce a fully functional application. By leveraging programming languages, we tried to build a robust web application that ensures smooth and efficient hotel operations.

General Conclusion:

This project successfully delivered a functional hotel reservation system based on room management, reservation management, and online reservation. Using PHP, HTML, CSS, JavaScript, and MySQL, we built a platform that serves four types of users: admin, accountant, director, and front-desk staff. Testing confirmed that all core features work correctly, including real-time room updates, double-booking prevention, and role-specific dashboards.

While notifications and online payment remain unfinished due to time limitations, the system is fully operational for manual reservation processing and administrative oversight.

The work accomplished during this internship demonstrates that a small development team can modernize hotel operations using standard web technologies, moving from paper-based methods to an integrated digital solution that improves both guest experience and staff efficiency.

For future work, we intend to complete the currently missing features and explore the possibility of extending the platform to additional domains, such as human resource management, restaurant operations, e-payment, or AI-driven services.

Bibliography

[1] Booking.com.Partner Portal. 2025. url: https://partner.booking.com (visited on 15/03/2026)

[2] Michael Kofler. “What is MySQL” In:MySQL. Spring,2001,pp.3-19. (visited on 03/04/2020)

[3] W3C. HTML5 Specification. Word wide web consortium,2024url: https://www.w3.org/html

[4] W3C.CSS3 Specification. World wide web consortium, 2024. url: https://www.w3.org/Style/CSS/ (visited on 20/04/2026)

[5] MDN Web Docs. JavaScript Documentation. Mozilla Developer Network, 2024. url : https://developer.mozilla.org/en-US/docs/Web/JavaScript (visited on 20/04/2026)

[6] The PHP Group.PHP: Hypertext Preprocessor official Documentation. 2014.url: https://www.php.net/docs.php (visited on 20/04/2026)

[7] Oracle corporation. MYSQL Official DOCUMENTATION. 2024.url: https://dev.mysql.com/doc/ (visited on 15/03/2026)

[8] Apache Friends. WAMP—Local Development Environment. 2024.url: https://www.apachefriends.org (visited on 20/04/2026)

[9] Microsoft.Visual Studio Code Documentation. 2024.url : https://code.visualstudio.com/docs (visited on 20/04/2026)

[10] GitHub Inc. GitHub Documentation—version control platform. 2024.url : https://docs.github.com (visited on 15/03/2026)

[11] Wikipedia contributors. Unified Modeling Language. Wikipedia, 2025. url: https://en.wikipedia.org/wiki/Unified_Modeling_Language (visited on 15/03/2026)

[13]Expedia Group. *Expedia Partner Solutions*. 2024. URL: https://www.expediapartnersolutions.com (visited on 15/03/2026)

[14] Mews System.Mews PMS—Hotel Management Software.2024.url: https://www.mews.com (visited on 15/03/2026)

[15] Airbnb. *Airbnb Official Website*. 2024. URL: https://www.airbnb.com (visited on 15/03/2026)

[16]Hotels. com. *Hotels.com— Hotel Booking Platform*. 2024. URL: https://www.hotels.com (visited on 15/03/2026)

[17] Oracle Hospitality. *Opera Property Management System*. 2024. URL: https://www.oracle.com/hospitality/ (visited on 15/03/2026)

[18] Cloudbeds. *Cloudbeds — Hotel Management Software*. 2024. URL: https://www.cloudbeds.com (visited on 15/03/2026)