

**UNIVERSITY OF ECONOMICS AND LAW
FACULTY OF INFORMATION SYSTEMS**



FINAL PROJECT

INFORMATION SYSTEM ANALYSIS & DESIGN COURSE

CLASS: 241IS4202

BOOKING HOTEL WEBSITE

Instructor: Le Thi Kim Hien, Ph.D.

Group 1's members:

Le Thuy Trang - K224060849

Tran Nguyen Nhat Linh - K224060839

Doan Thuy Trang – K224060848

Ho Khai Yen – K224060854

Ngo Thanh Van - K224060852

Ho Chi Minh, 16 November 2024

MEMBERS OF GROUP 1

No.	Full name	Student ID	Point / 10 (Individual Contribution)	Signature
1	Le Thuy Trang	K224060849	10/10	
2	Tran Nguyen Nhat Linh	K224060839	10/10	
3	Doan Thuy Trang	K224060848	10/10	
4	Ho Khai Yen	K224060854	10/10	
5	Ngo Thanh Van	K224060852	10/10	

ACKNOWLEDGEMENTS

First and foremost, we would like to express our sincere gratitude to the University of Economics and Law for incorporating the "Information System Analysis & Design Course" course into the curriculum. This course not only provided us with specialized knowledge but also broadened our understanding of the role and importance of this field in the digital economy.

In particular, we would like to extend our heartfelt thanks to Ms. Le Thi Kim Hien, who has been dedicated in her guidance and always willing to share valuable insights. Thanks to her passionate lectures and practical exercises, we have had the opportunity to apply theoretical knowledge to real-world tasks, helping us understand specific problem-solving approaches.

However, due to limitations in time and knowledge, our report certainly contains some shortcomings. We sincerely hope to receive constructive feedback and guidance from Ms. Hien to help us improve our work.

Group 1

COMMITMENT

We would like to affirm that the data used for the analysis in this thesis were carefully selected from transparent sources and thoroughly documented. The research team has also fully complied with current regulations on research and data publication to ensure the transparency and credibility of this study.

The research results presented in this report are the product of continuous effort by the research team, grounded in a strong commitment to honesty, objectivity, and enthusiasm for applying research to the real-world context of Vietnam. Each step in the research process was conducted with the highest level of integrity and strict adherence to ethical principles.

We would particularly like to emphasize the novelty and significance of the research findings in this thesis. These are not merely research outcomes but represent unique and valuable contributions to knowledge in this field, never before published or disseminated in any other research.

Ho Chi Minh City, 2024

Group 1

SUMMARY

With constantly shifting customer demands and increasing market competition, hotels need to rely on data-driven strategies to improve operational efficiency and enhance booking rates.

This online hotel booking application utilizes advanced technologies such as Angular, MongoDB, Node.js, and HTML to provide users with a quick, convenient, and personalized booking experience. The platform features a user-friendly interface that allows customers to search and book rooms according to their preferences in just a few simple steps. Diverse search filters enable users to easily compare and select rooms based on criteria such as location, price, and amenities. On the technical side, the back-end of the application is built on Node.js and MongoDB, ensuring scalability and efficient data management, especially when handling large transaction volumes and simultaneous access. Angular is used for front-end development, creating a modern, flexible user interface with smooth interactions. HTML helps maintain a consistent website structure and optimizes the user experience across multiple devices, from desktops to mobile phones. In addition to meeting booking needs, the application integrates intelligent data analytics and reporting tools, enabling hotels to gain deeper insights into customer behavior and preferences, thereby optimizing business strategies. Through this platform, hotels can maximize occupancy rates, improve service quality, and provide customers with a fast and efficient booking journey.

To satisfy all objectives above for an optimized online hotel booking application, the team applied all needed diagram such as BPMN, Use case diagram, Sequence diagram, Activity diagram, Class diagram to clarify the business process in terms of technical aspects.

TABLE OF CONTENT

MEMBERS OF GROUP 1	i
ACKNOWLEDGEMENTS	ii
COMMITMENT	iii
SUMMARY	iv
TABLE OF CONTENT	v
LIST OF FIGURES.....	ix
LIST OF TABLES.....	xii
LIST OF ABBREVIATIONS.....	xiv
CHAPTER 1: PROJECT OVERVIEW.....	1
1.1. The introduction of Dé Luna Hotel	1
1.1.1. Brand name	1
1.1.2. Brand story	1
1.1.3. Vision, mission and goals	1
1.2. Deliverables	2
1.3. Overview of information system	3
1.4. Business problems statement/Business needs	4
1.5. Propose solutions	5
1.6. Assumptions	5
1.7. System impacted.....	6
1.8. Introduction to system's methodologies.....	7
CHAPTER 2: FACT-FINDING AND REQUIREMENTS MODELING	11
2.1. Introduction to techniques applied for Fact-finding and Requirements modeling	11
2.1.1. Fact-finding Techniques	11
2.1.2. Requirements Modeling Techniques	14
2.2. Deliverables	15
2.3. Current system and system architecture/diagrams (As-is business)	17
2.4. Business Requirement Documents	19
2.5. Systems Requirements Documents	23
2.5.1. Functional Requirements	23

2.5.2. Non-Functional Requirements	24
CHAPTER 3: DATA AND PROCESS MODELING	27
3.1. Overview and purposes	27
3.2. Deliverables	28
3.3. Business process modeling and notation (BPMN)	29
3.3.1. Process overview.....	29
3.3.2. Login/Sign up process.....	30
3.3.3. Update the booking (admin)	31
3.4. Requirement modeling by DFD (Data Flow Diagram).....	32
3.4.1. Data sources, data flows, data store, data destinations and process	32
3.4.2. Context diagram – Context level	33
3.4.3. Level 0 of DFD	35
3.4.4. Level 1 of DFD	37
3.5. Data Dictionary.....	42
3.5.1. Data flow	42
3.5.2. Data store	43
3.5.3. Entity	45
3.5.4. Process.....	46
CHAPTER 4: OBJECT MODELING.....	48
4.1. Overview and purposes	48
4.2. Deliverables	48
4.3. Scenarios, Models and Functionality	49
4.3.1. Use Case Diagram.....	49
4.3.2. Use Case Descriptions (User story)	50
4.4. Class Diagram.....	80
4.5. Sequence Diagram.....	81
4.6. Activity Diagram	85
CHAPTER 5: USER INTERFACE DESIGN AND DATA DESIGN	87
5.1. User interface design	87
5.1.1. Overview and purposes	87
5.1.2. Deliverables.....	87

5.1.3. Sitemap.....	88
5.1.4. Screen Mockups and Prototype.....	89
5.2. Validation	108
5.2.1. User input	108
5.2.2. Admin Input	109
5.3. Database design	109
5.3.1. Overview and purpose	109
5.3.2. Deliverables.....	110
5.3.3. Identify collection, field, data types.....	111
CHAPTER 6: SYSTEMS IMPLEMENTATION	115
6.1. Coding	115
6.2. Programming languages and tools.....	120
6.2.1. The Choice of the Language	120
6.2.2. Programming Tools.....	120
6.3. Testing	127
6.3.1. Overview	127
6.3.2. Test Scenarios	127
6.3.3. Test results	128
6.4. Deployment	132
6.4.1. Steps in the deployment process	132
6.4.2. Deployment diagram.....	134
6.5. User training	135
CHAPTER 7: SYSTEMS SECURITY AND SUPPORT	138
7.1. Threat Prediction and Analysis	138
7.2. Update and Maintenance System	141
7.2.1. Maintenance process	141
7.2.2. Update and Maintenance Plan.....	142
CHAPTER 8: PROJECT PLAN.....	146
8.1. Work Breakdown Structure	146
8.2. Pert Chart.....	148
8.3. Gantt Chart	149

CHAPTER 9: CONCLUSION	150
9.1. Result	150
9.2. Lessons Learnt	151
REFERENCES	156

LIST OF FIGURES

Figure 3. 1. Business process modeling and notation - Process overview (Source: Authors)	29
Figure 3. 2. Business process modeling and notation - Login/ Sign up process (Source: Authors)	30
Figure 3. 3. Business process modeling and notation - Update the booking (Source: Authors)	31
Figure 3. 4. Context Diagram (Source: Authors)	34
Figure 3. 5. Level 0 of DFD (Source: Authors)	35
Figure 3. 6. DFD level 1 of process 1 (Source: Authors)	37
Figure 3. 7. DFD level 1 of process 2 (Source: Authors)	39
Figure 3. 8. DFD level 1 of process 3 (Source: Authors)	41
Figure 3. 9. Data flow (Source: Authors)	43
Figure 3. 10. Data store (Source: Authors)	45
Figure 3. 11. Entity (Source: Authors)	46
Figure 3. 12. Process (Source: Authors)	47
Figure 4. 1. Use Case Diagram (Source: Authors)	50
Figure 4. 2. Class Diagram (Source: Authors)	80
Figure 4. 3. Sequence Diagram for Register (Source: Authors)	81
Figure 4. 4. Sequence Diagram for Login (Source: Authors)	82
Figure 4. 5. Sequence Diagram for Booking (Source: Authors)	83
Figure 4. 6. Activity Diagram for Booking (Source: Authors)	85
Figure 5. 1. Sitemap (Source: Authors)	88
Figure 5. 2. Header interface of the website (Source: Authors)	89
Figure 5. 3. Footer interface of the website (Source: Authors)	90
Figure 5. 4. The login page interface of the website (Source: Authors)	91
Figure 5. 5. The password reset interface of the website (Source: Authors)	91
Figure 5. 6. The signup page interface of the website (Source: Authors)	92
Figure 5. 7. Account & Edit Information Tab (Source: Authors)	92
Figure 5. 8. Home interface of the website (Source: Authors)	93

Figure 5. 9. About us interface of the website (Source: Authors).....	94
Figure 5. 10. Room interface of the website (Source: Authors).....	95
Figure 5. 11. Room details (Source: Authors).....	96
Figure 5. 12. Booking list (Source: Authors)	96
Figure 5. 13. Confirm information (Source: Authors)	97
Figure 5. 14. Payment (Source: Authors)	98
Figure 5. 15. Dinning interface of the website (Source: Authors)	99
Figure 5. 16. Dining detail interface of the website (Source: Authors)	100
Figure 5. 17. Meeting interface of the website (Source: Authors)	101
Figure 5. 18. Service interface of the website (Source: Authors)	102
Figure 5. 19. The admin login page interface (Source: Authors).....	103
Figure 5. 20. Admin Customer Management Page Interface (Source: Authors)	104
Figure 5. 21. Edit Customer Information (Source: Authors)	104
Figure 5. 22. Delete Customer Information (Source: Authors).....	105
Figure 5. 23. View Available Rooms List (Source: Authors)	105
Figure 5. 24. Edit Room Details (Source: Authors)	106
Figure 5. 25. Delete Room (Source: Authors).....	106
Figure 5. 26. Booking Details (Source: Authors).....	107
Figure 5. 27. Edit Status Booking (Source: Authors).....	107
Figure 5. 28. Admin logout interface (Source: Authors)	108
Figure 6. 1. Router Search available room (Source: Authors)	117
Figure 6. 2. Controller searchRoomAvaiable function (Source: Authors)	118
Figure 6. 3. Middleware verify admin (Source: Authors)	120
Figure 6. 4. Logo Angular (Source: Internet).....	121
Figure 6. 5. Advantages of Angular (Source: Internet)	121
Figure 6. 6. Disadvantages of Angular (Source: Internet)	123
Figure 6. 7. Logo Mongodb (Source: Internet)	124
Figure 6. 8. Logo NodeJS (Source: Internet)	126
Figure 6. . Deployment diagram (Source: Authors)	135
Figure 7. 1. Maintenance process (Source: Authors)	142
Figure 8. 1. Work Breakdown Structure (Source: Authors).....	146

Figure 8. 2. Pert Chart (Source: Authors)..... 148

Figure 8. 3. Gantt Chart (Link) (Source: Authors)..... 149

LIST OF TABLES

Table 1. 1. Systems Development Methods (Source: Tilley, S., & Rosenblatt, H. J., 2017)	7
Table 2. 1. Customer interview information (Source: Authors).....	12
Table 2. 2. Customer interview questions (Source: Authors)	12
Table 2. 3. Employee interview information (Source: Authors)	13
Table 2. 4. Employee interview questions (Source: Authors).....	13
Table 2.5. Business requirements (Source: Authors)	20
Table 2.6. Key stakeholders (Source: Authors).....	21
Table 2.7. Project Constraints (Source: Authors).....	22
Table 2.8. Cost-Benefits Analysis (Source: Authors)	22
Table 2.8. Functional requirements (Source: Authors)	23
Table 2.9. Non-functional requirements (Source: Authors).....	24
Table 4. 1. Use case specification of Subscribe for promotion (Source: Authors)	50
Table 4. 2. Use case specification of View Room (Source: Authors).....	53
Table 4. 3. Use case of Register (Source: Authors)	54
Table 4. 4. Use case specification of Login (Source: Authors).....	57
Table 4. 5. Use case specification of Forgot Password (Source: Authors)	59
Table 4. 6. Use case specification of Search available room (Source: Authors)	61
Table 4. 7. Use case specification of Booking (Source: Authors)	63
Table 4. 8. Use case specification of Choose Quantity (Source: Authors)	64
Table 4. 9. Use case specification of Fill Contact Information (Source: Authors)	66
Table 4. 10. Use case specification of Confirm Booking (Source: Authors)	67
Table 4. 11. Use case specification of booking cancel (Source: Authors).....	68
Table 4. 12. Use case specification of payment (Source: Authors)	69
Table 4. 13. Use case specification of Logout (Source: Authors).....	72
Table 4. 14. Use case Manage Users (Source: Authors)	73
Table 4. 15. Use case Manage Room (Source: Authors)	75
Table 4. 16. Use case Manage Bookings (Source: Authors).....	77
Table 5. 1. Validation for user input (Source: Authors).....	108

Table 5. 2. Validation for user input (Source: Authors).....	109
Table 5. 3. Description of the database (Source: Authors)	111
Table 6. 1. Coding Search available room description (Source: Authors).....	115
Table 6. 2. Verify admin function description (Source: Authors).....	118
Table 6. 3. Language used (Source: Authors)	120
Table 6. 4. Scenarios for each type of testing (Source: Authors).....	127
Table 6. 5. Issues discovered in the hotel's website (Source: Authors)	128
Table 6. 6. Issues detected in Admin's hotel management page (Source: Authors) ..	130
Table 6. 7. Customer training for hotel booking website Dé Luna Hotel (Source: Authors).....	135
Table 6. 8. Admin training for hotel booking management website Dé Luna Hotel (Source: Authors)	136
Table 7. 1. Threat Prediction and Analysis (Source: Authors)	138
Table 7. 2. Detailed plan (Source: Authors).....	143
Table 7. 3. Maintenance and update schedule (Source: Authors)	145
Table 9. 1: Result for Booking Hotel Website Project (Source: Authors)	150
Table 9. 2. Lessons Learned and Improvements for Booking Hotel Website Project (Source: Authors)	151
Table 9. 3. Success Factors and Best Practices (Source: Authors)	153

LIST OF ABBREVIATIONS

No.	Abbreviations	Meanings
1	AD	Admin
2	AD	Adaptability
3	AP	Accuracy and Precision
4	BPMN	Business Process Model and Notation
5	BSON	Binary Javascript Object Notation
6	CRM	Customer Relationship Management
7	CRM	Customer Relationship Management
8	DFD	Data Flow Diagrams
9	EF	Exceptional Flow
10	ERD	Entity-Relationship Diagrams
11	HMIS	Hotel management information systems
12	IO	Information-Oriented Requirements
13	JSON	JavaScript Object Notation
14	OOA	Object Oriented Analysis
15	PC	Platform Constraints
16	PCI-DSS	Payment Card Industry Data Security Standard
17	PF	Primary Flow
18	PMS	Property Management System
19	PO	Process-Oriented Requirements
20	PR	Performance Requirements
21	PUI	Project Unique Identifier
22	SC	Security
23	SDLC	Software Development Life Cycle
24	SMS	Short Message Services
25	SQL	Structured Query Language
26	UI/UX	User experience/ user interface
27	UML	Unified Modeling Language

CHAPTER 1: PROJECT OVERVIEW

1.1. The introduction of Dé Luna Hotel

1.1.1. Brand name

The brand name "Dé Luna Hotel" draws inspiration from both Latin and French influences, where "Luna" translates to "Moon," and "Dé" carries a refined tone, evoking a sense of belonging or connection to the Moon. In astrology, the Moon represents human emotions, moods, and social interactions, reflecting our perception of ourselves. In French culture, moonlight often evokes a sense of romance, mystery, and elegance, qualities frequently found in poetry and literature. With this symbolic meaning, Dé Luna Hotel aims to offer a comfortable, poetic, and distinctly French atmosphere, providing guests with a tranquil space where they can experience peace, freedom, and the charm of a moonlit night.

1.1.2. Brand story

Dé Luna Hotel is a hotel with a French romantic style. Inspired by the peaceful, lyrical beauty of moonlight, Dé Luna Hotel aims to bring a healing resort space to customers. With the name "Dé Luna", the core value that the hotel wants to convey is to create a comfortable, poetic atmosphere like the moonlight falling on a soft dark night. Here, each customer has the opportunity to rediscover themselves, fully feel their emotions and soul. Away from the hustle and bustle of life outside, enjoy the fresh air. Dé Luna Hotel promises to become an ideal destination for visitors to freely explore themselves amid a poetic, romantic and lyrical atmosphere like moonlight.

1.1.3. Vision, mission and goals

1.1.3.1. Vision

Dé Luna Hotel strives to become one of the leading resorts with a romantic French style, offering a healing retreat where each customer can find peace and rediscover themselves.

1.1.3.2. Mission

To create a retreat space that brings comfort, tranquility, and genuine emotions to every customer. We are committed to delivering high-end lodging experiences, combined with professional and dedicated services, helping customers deeply feel the serenity and joy of life.

1.1.3.3. Goals

To become the favorite hotel for both domestic and international guests, especially those who seek peace and tranquility.

To achieve customer satisfaction through dedicated customer service and a high-end retreat space.

To secure a leading position in the resort hotel industry and enhance the brand's presence in the tourism market.

1.2. Deliverables

Project Plan: Detailed document describing how the project will be implemented, including goals, scope, schedule, budget, key milestones, and required resources.

Requirements Analysis: Document that analyzes requirements from stakeholders, identifying features and functions needed on the booking website.

User interface design (UI/UX Design): Interface and user experience designs for the website, ensuring ease of use and optimization for users.

Development and programming: The process of coding and developing website features.

Testing: Check the entire website system to ensure quality, security, and stability.

Website Deployment (Deployment): The process of deploying a website to a production environment and making it live.

Maintenance and Support: Post-deployment maintenance activities, including bug fixes, feature updates, and user support.

1.3. Overview of information system

Hotel management information systems (HMIS) are pivotal for operational efficiency in hospitality. Integrating various functionalities, these systems manage bookings, customer interactions, and backend operations seamlessly, offering a comprehensive suite for modern hotel management needs. From online bookings to room services, HMIS covers all bases, ensuring smooth operations across departments. (Bali, O., 2024)

Real-Time Data Access and Management

A key feature of contemporary HMIS is its capacity to deliver real-time access to vital data, including guest contact details, online bookings, and room service requests. This immediacy allows for instant updates across all platforms, enabling staff to promptly address guest needs and manage operations efficiently.

Integration with Daily Operations

HMIS integrates smoothly with a hotel's everyday functions, encompassing everything from housekeeping and maintenance schedules to sales and marketing efforts. This integration streamlines processes, minimizes errors, and ensures efficient operations. For example, scheduling tools can alert housekeeping staff about last-minute room changes, while marketing solutions can offer personalized promotions based on guest preferences.

Property Management System (PMS) Integration

The HMIS connects with the hotel's internal Property Management System (PMS) to synchronize critical functions such as room availability, guest check-in/check-out procedures, and housekeeping tasks. This real-time status update helps prevent overbooking and enhances guest management. The PMS serves as a centralized hub for

managing bookings, billing, guest interactions, and more, simplifying oversight for managers by consolidating all operational facets into a single interface.

Payment Gateway Integration

The system incorporates secure payment options that allow guests to pay via credit/debit cards, mobile wallets, and other digital payment methods. It adheres to PCI-DSS standards for secure payment data handling and offers flexibility with prepayment, post-payment, or split payment options for group bookings.

Customer Relationship Management (CRM)

The CRM component maintains a comprehensive database of customer information, including booking history and preferences, enabling personalized guest experiences. It supports targeted marketing campaigns and loyalty programs by offering tailored discounts to returning customers. Additionally, customer support staff can access relevant information to effectively address inquiries and resolve issues.

When a guest visits the online booking website, they input their travel dates and apply filters for room types and amenities. The reservation system then displays available rooms along with rates and details. After selecting a room and completing the payment process, the system confirms the booking while updating availability in real time across all platforms. The guest receives an email confirmation followed by a reminder prior to their stay. During check-in, front desk staff utilize the PMS to retrieve booking details, ensuring a smooth check-in experience. Subsequently, the CRM records the guest's preferences for future visits, allowing for personalized offers tailored to their needs.

1.4. Business problems statement/Business needs

Currently, the hotel booking process often takes place directly at the reception desk or via phone or online platforms for guests who book in advance. This is quite inconvenient for customers, especially those who live far away. Furthermore, when customers need to consult rooms (prices, room status, services, etc.), they also spend a lot of time receiving consulting information from the customer care department, which affects the customer experience.

1.5. Propose solutions

Developing website

Create a website that allows customers to easily book rooms remotely, regardless of time and location, include some features such as viewing rooms, booking rooms, and online payment. Which helps customers easily book rooms and access detailed information quickly and conveniently

Automated notifications and reminders via email/SMS

Send notifications via email to customers when there are price changes and special promotions.

Streamlined booking processes make it easier for customers to reserve rooms, significantly enhancing their overall experience. By utilizing technology for customer service, the operational burden on hotel staff is reduced, allowing them to focus on providing quality service. These solutions not only improve customer satisfaction but also lead to increased bookings and positive reviews, ultimately benefiting the hotel's reputation and profitability.

1.6. Assumptions

User Assumption

All users are expected to be capable of using technological devices such as smartphones or tablets. To use the service, users will create an account on the hotel booking platform using a valid email address, from which they will receive information and booking confirmations. They should also have a basic understanding of how to use the booking application. Users are assumed to have a stable Internet connection to carry out online booking transactions.

Database and Security System

The hotel booking platform is assumed to have a fast, secure, and reliable transaction processing system. The system will be capable of handling multiple booking requests without encountering issues. Additionally, the platform will have a well-structured

database to store information and ensure that information is regularly updated. The website will also implement strong security measures to protect customers' personal and payment information

Operational Process and Customer Support

The customer service team will be available to assist customers with issues related to bookings, cancellations, and changes to booking information. Customers will be notified of their booking status via email or through notifications within the application, ensuring they are always updated. Furthermore, users will be able to easily interact with the booking platform through features such as room search, selecting additional services, and online payment.

User Experience

All users will accept the terms and conditions of using the booking service, including the processing of personal information and requests for access to features within the website. Finally, customers are expected to provide feedback about their booking experience, which will help improve services and enhance quality in the future.

1.7. System impacted

Hotel Booking Platform

The current booking system will need significant updates to incorporate online booking features, allowing customers to reserve rooms through the hotel's website. This change will include the development of an intuitive interface that supports searching, viewing room details (pricing, availability), and finalizing bookings.

Database and Security System

The hotel's database infrastructure must be expanded to handle an increase in data from online room reservations, customer profiles, and transaction records. Enhanced data storage capacity and robust security measures are essential to safeguard customer information, including personal details and payment information.

Operational Process and Customer Support

The operational workflow will be revised to ensure coordination between in-person and online booking systems. This involves updating check-in procedures and synchronizing online and in-person reservations to prevent double bookings. Staff will require training in managing the new integrated system to maintain efficiency and ensure room status updates are accurately reflected across platforms. This includes training to address issues related to digital booking processes, assisting customers with account management, and providing remote support for any booking issues. The support process will need adjustments to manage a potential increase in queries related to online transactions and system navigation.

1.8. Introduction to system's methodologies

Table 1. 1. Systems Development Methods (Source: Tilley, S., & Rosenblatt, H. J., 2017)

	Structured Analysis	Object-oriented Analysis	Agile Methods
Description	Represents the system in terms of data and the processes that act upon that data. System development is organized into phases, with deliverables and milestones to measure progress. The waterfall model typically consists of five phases: requirements, design,	Views the system in terms of objects that combine data and processes. The objects represent actual people, things, transactions, and events. Compared to structural analysis, O-O phases tend to be more interactive. Can use the waterfall model or a model that	Stresses intense team-based effort. Breaks develop into cycles, or iterations, that add functionality. Each cycle is designed, built, and tested in an ongoing process. Attempts to reduce major risks by incremental steps in short time intervals.

	construction, testing, and maintenance & evolution. Iteration is possible among the phases.	stresses greater iteration.	
Modeling tools	Data flow diagrams (DFDs) and process descriptions, which are described in Chapter 5. Also, business process modeling.	Various object-oriented diagrams depict system actors, methods, and messages, which are described in Chapter 6. Also, business process modeling	Tools that enhance communication, such as collaborative software, brainstorming, and whiteboards. Business process modeling works well with agile methods.
Advantages	Traditional method that has been very popular over time. Relies heavily on written documentation. Frequent phase iteration can provide flexibility comparable to other methods. Well-suited to traditional project management tools and techniques	Integrates easily with object-oriented programming languages. Code is modular and reusable, which can reduce cost and development time. Easy to maintain and expand because new objects can be created using inherited properties	Very flexible and efficient in dealing with change. Stresses team interaction and reflects a set of community-based values. Frequent deliverables constantly validate the project and reduce risk
Disadvantages	Changes can be costly, especially in later phases.	Somewhat newer method might be less familiar to	Team members need a high level of technical and

	<p>Requirements are defined early and can change during development. Users might not be able to describe their needs until they can see examples of features and functions.</p>	<p>development team members. Interaction of objects and classes can be complex in larger systems.</p>	<p>communications skills. Lack of structure and documentation can introduce risk factors. Overall project might be subject to scope change as user requirements change</p>
<p>Application in Hotel Booking System</p>	<p>SDLC can be used to ensure each system component—such as booking management, payment processing, and customer relationship management—is thoroughly analyzed and documented before development begins. The method is particularly suitable if the system requirements are stable and unlikely to change significantly, as is often the case for core hotel operations.</p>	<p>OOA is well-suited for the hotel booking domain, as the system can be structured around objects like "Guest," "Booking," "Room," and "Payment." This approach simplifies the handling of complex relationships (such as between reservations, guests, and rooms) and allows for efficient code maintenance as new features or enhancements are introduced.</p>	<p>Agile can be advantageous if the hotel requires frequent updates to adapt to changing market demands, such as adding seasonal promotions, introducing new payment methods, or customizing user experiences. By delivering the system in iterative cycles, developers can prioritize and release core features first—such as room booking and availability checks—while continually enhancing the system</p>

			based on feedback from both users and hotel management.
--	--	--	---

CHAPTER 2: FACT-FINDING AND REQUIREMENTS MODELING

2.1. Introduction to techniques applied for Fact-finding and Requirements modeling

When developing an online hotel booking system - Dé Luna Hotel, effective fact-finding and requirements modeling techniques are crucial to ensure that the system aligns with the needs of both users and hotel administrators. These techniques help identify system functionalities, gather insights from stakeholders, and translate business objectives into technical requirements.

2.1.1. Fact-finding Techniques

2.1.1.1. Overall fact-finding

Fact-finding involves collecting detailed information about the project from various sources to define the requirements of the system accurately (UKEssays, 2018).

Interviews: Direct interviews with hotel staff, management, and potential users provide insights into their expectations, challenges, and needs. By interviewing front-desk personnel, administrators, and customers, we can better understand desired features like booking workflows, room management, and secure payment processes.

Questionnaires and Surveys: Distributing surveys to both current hotel guests and employees offers a broad perspective on user needs. These tools help in gathering information about features such as payment preferences, reservation processes, and user interface expectations.

Observation: Observing current hotel booking practices and front-desk operations helps us see real-world processes in action. Watching hotel staff interact with current systems can reveal pain points and highlight features that could improve workflow.

Document Analysis: Reviewing existing documents like booking records, guest feedback forms, and transaction logs reveals recurring issues, seasonal booking trends, and room turnover rates. This analysis is helpful for defining system needs like room availability updates, check-in/check-out workflows, and reservation histories.

Workshops: Collaborative workshops bring together hotel managers, IT staff, and stakeholders to discuss the objectives and limitations of the system. Workshops facilitate brainstorming and foster alignment on required functionalities, such as CRM integration, dynamic pricing, and channel management.

2.1.1.2. Detail fact-finding interview plan

This is fact-finding interview plan for Website Dé Luna Hotel:

Customer Interview:

- Objective: Identify customer expectations and preferences for the booking experience.

Table 2. 1. Customer interview information (Source: Authors)

Interview Details	Description
Interviewee	Hotel Guests (both new and returning)
Interviewer	Business Analyst
Duration	30 minutes
Mode	In-person or online survey

- Questions for customers

Table 2. 2. Customer interview questions (Source: Authors)

Category	Questions
Booking	1. How do you usually book hotel rooms (online, phone, walk-in)?
Experience	

	2. What do you find most frustrating about your current booking process?
Features & Preferences	3. What features do you look for in a hotel booking website (e.g., search filters, reviews)?
	4. Would you prefer personalized recommendations based on past stays?
Payment	5. What payment methods do you prefer (credit card, e-wallets, etc.)?
Support	6. How important is 24/7 customer support to you?
Notifications	7. Would you like to receive booking updates and promotions via email or SMS?

Employee Interview (Hotel Staff):

- Objective: Understand the operational challenges, current processes and desired system functionalities.

Table 2. 3. Employee interview information (Source: Authors)

Interview Details	Description
Interviewee	Front Desk Staff, Admin Staff
Interviewer	Business Analyst
Duration	45 minutes
Mode	In-person or virtual

- Questions for employees

Table 2. 4. Employee interview questions (Source: Authors)

Category	Questions
Current Process	1. How do you currently handle room bookings (walk-ins, phone, emails)?

	2. What challenges do you face in managing bookings during peak times?
Desired Features	3. What features would make your job easier (e.g., automated availability updates)?
	4. Would a chatbot be useful for handling common customer queries?
Data & Reporting	5. What type of reports do you need for managing bookings and customer data?
Integration	6. What systems (e.g., PMS, payment gateways) should the platform integrate with?
Security	7. What security concerns do you have regarding customer data?

After conducting the interviews, the insights will be summarized into a comprehensive Requirements Document, ensuring the platform aligns with both operational needs and customer expectations.

2.1.2. Requirements Modeling Techniques

Requirements modeling translates fact-finding insights into structured system requirements. This process focuses on defining both functional (what the system does) and non-functional (system quality and performance) requirements.

Use Case Diagrams: Use case diagrams identify system actors and their interactions with different functions. For an online hotel booking system, use cases could include “Make a Reservation”, “Process Payment”, “Update Room Availability”, “Generate Booking Reports”, etc. These diagrams help visualize user interactions and ensure coverage of key functionalities.

Data Flow Diagrams (DFD): DFDs map out how information flows within the system, illustrating connections between users, data storage and processing steps. For example, a DFD might show how booking information flows from the guest interface to the management system, ensuring accurate availability tracking.

Entity-Relationship Diagrams (ERD): ERDs define data entities (such as customers, bookings, rooms) and the relationships between them, providing a clear structure for database design. This modeling is critical for understanding data dependencies and ensuring that customer details, booking history and payment records are stored securely.

Functional Decomposition: This technique breaks down complex system functions into smaller, manageable components. For instance, the “Manage Rooms” function could be divided into sub-functions like “Create Room”, “Modify Room” and “Delete Room”. This decomposition helps in assigning tasks to development teams and clarifies system scope.

User Stories and Scenarios: User stories capture specific requirements from the user’s perspective (e.g., “As a guest, I want to view room availability in real time.”). These stories clarify user needs and aid in developing user-centered features. Scenarios based on these stories help illustrate real-world usage cases and guide design decisions.

Prototyping: Building an interactive prototype of the booking system offers a tangible representation of the user interface and functionality. Prototypes allow stakeholders to visualize the booking process, payment options and account management features, facilitating early feedback and iterative improvement.

2.2. Deliverables

Deliverables, which refer to the specific outputs, products, or results that a project intends to produce and deliver to its stakeholders, are crucial to project management.

Internal Deliverables:

- Project Plan: Details of implementation steps, timeline, work assignments and necessary resources for the project.
- Status Reports: Periodic reports to monitor progress and project implementation.
- Risk Assessments: Potential risk assessment report and risk mitigation plan throughout the website development process.

- Design Specifications: Detailed requirements for the interface and functionality of the website.
- Test Plans and Reports: Test plans for website features and report test results.
- Project Milestones: Important milestones to track project development progress.

External Deliverables:

- Website: The final product – the hotel booking website, is developed and delivered to the customer or end user.
- User Documentation: Documentation guiding how to use the website for end users, including instructions for booking, managing user accounts, and making payments.

Process Deliverables

- Prototypes: Samples of the user interface (UI) or website functionality, to get feedback and adjustments before official development.
- Design Mockups: Detailed designs for the website's user interface.
- Beta Version of Website: A test version of the website is provided to a group of test users to receive feedback.
- Testing Reports: Reports on testing website functions (testing booking, payment, account management, etc.).

Product Deliverables

- Final Website (Complete Website): Complete hotel booking website with fully operational functions such as room search, booking, payment, and hotel reviews.
- Payment Integration: Online payment function integrated into the website, allowing customers to pay when booking.
- Booking Management System: Backend system that helps manage hotel reservations, room availability, and reservation confirmations.
- Mobile Compatibility: Ensure the website can be accessed and used easily on mobile devices (responsive design).

- Admin Management System: Management interface for administrators to track and manage reservations, customers, and promotions.

Tangible Deliverables:

- Website Source Code: Website source code, HTML, CSS, JavaScript files, and necessary resources to operate the website.
- Server Setup: The web server is configured and deployed to host the website.

Intangible Deliverables:

- Software Features: Features such as room search, booking, online payment.
- User Experience: Improvements to the interface and user experience (UX) for the website, helping users easily operate and complete bookings.
- Training Program: Training program for employees or website administrators on how to use the administration system and other support tools.
- Business Process Development: Build processes related to processing reservations, payments and customer support.

2.3. Current system and system architecture/diagrams (As-is business)

Current System:

The current hotel booking system operates based on manual methods such as phone calls, emails, and in-person bookings at the front desk. When a customer wants to book a room, they must call or email to inquire about room availability, pricing, and make a reservation. Front desk staff handle each request, check room availability in the internal system, record the booking, and send a confirmation to the customer via email or phone. All these steps are done manually, which creates a high risk of errors, especially during peak hours or when there is a large volume of booking requests.

The payment process is also entirely manual. Customers typically pay when they check in or through bank transfer, and front desk staff must manually reconcile payment transactions. The current system lacks an online booking platform, preventing customers from checking room availability, making reservations, or paying in advance

through the internet. This not only causes inconvenience for customers but also reduces staff efficiency, increases the risk of overload during peak seasons, and makes it difficult to scale operations.

High-Level System Architecture:

The upgraded hotel booking website system will adopt a 3-tier software architecture, consisting of three main tiers: Front-End, Middleware, and Back-End. This architecture provides a comprehensive improvement over the current system, allowing customers to easily search for, book, and pay for rooms online.

Front-End:

The Front-End tier provides an intuitive and user-friendly interface for customers to access the website. Here, customers can:

- Search for available rooms based on criteria such as check-in date, room type, price, and amenities.
- View images, detailed descriptions, and amenities of each room type.
- Make online bookings by selecting a room, adding additional services (e.g., breakfast, airport transfer), and completing the payment via a secure payment gateway.
- Manage their booking cart, track the booking process, and receive automatic notifications via email or SMS for booking confirmations and payment statuses.

Middleware:

The Middleware tier processes requests from the Front-End and connects to the Back-End to update information in real time. This layer provides services such as:

- **Data processing and routing:** It receives and processes room booking requests from the Front-End, checks room availability, and responds to the customer instantly.
- **Payment gateway integration:** Enables connection with external payment gateways (e.g., credit cards, digital wallets), ensuring that payments are processed securely and conveniently.

- **Security and user authentication:** Middleware integrates security features such as SSL encryption and two-factor authentication to protect customers' personal information during the transaction process.

Back-End:

The Back-End tier is responsible for managing and storing all critical system data, including:

- **Hotel data management:** Stores information about room types, room availability, pricing, and additional services.
- **Inventory and room status management:** The room availability data is updated in real time to ensure the system always displays accurate room statuses and minimizes the risk of overbooking.
- **Customer management:** Stores and secures customer information, including booking history, payment details, and special requests.
- **Reporting and analytics:** Generates regular reports on room status, revenue, and booking statistics to help hotel management monitor and optimize business operations.

2.4. Business Requirement Documents

Project name: Booking hotel website

Project manager: Le Thuy Trang

Date submitted: 17/11/2024

Document status: Done

Executive summary: The project of building an online hotel booking website aims to create a platform that allows customers to easily search and book hotel rooms online. The website will provide user support features such as searching for rooms by criteria, registering and logging in to an account, a reservation management system, sending email marketing campaigns, and integrating CRM to track history. and customer behavior.

Project Objectives:

- Build and deploy a complete online hotel booking website in 2 months.
- Integrate your customer management system (CRM) to improve customer retention and increase revenue from marketing campaigns.
- Ensure the user interface is easy to use, friendly and compatible on mobile devices.
- Optimize SEO to attract customers from search engines.
- 20% revenue growth within 1 year after the website went into operation.

Project Scope: The online hotel booking website will include features such as user-friendly interface design, flexible search and booking functions, and integrated secure online payment system. Users can register, log in and manage personal information and booking history. The CRM system will support tracking customer behavior, optimizing marketing and customer care strategies. In addition, the website will automatically send marketing emails and provide management pages for hotels to update rooms, room rates and track booking history.

Business requirements:

Table 2.5. Business requirements (Source: Authors)

Priority level	Critical level	Requirement description
1	High	User registration and login system, allowing users to store personal information and track booking history.
2	High	Search and booking function, support room selection by date, room type, number of guests and additional services.
3	High	Integrate online payment system (credit card, bank transfer, e-wallet).
4	Medium	Send emails to confirm bookings and notify customers about promotions and discounts.

5	Medium	Integrate CRM system to track booking history and user behavior, serving marketing and customer care strategies.
6	Low	Customer review and feedback function on room and service quality.
7	Low	The system automatically notifies customers about special offers and promotions via email or SMS.
8	Medium	Integrated advanced search engine, allowing customers to filter results by criteria such as price, room type, amenities, and previous customer reviews.
9	Low	Multilingual support for the website, helping international users to access and use the service easily.
10	Low	Provide a dashboard for administrators to monitor room status, revenue, and financial reports.

Key stakeholders:

Table 2.6. Key stakeholders (Source: Authors)

Name	Job role	Duties
Le Thuy Trang	Project manager	Is the bridge between the business part and the technical part of the project - the link between the client and the development team by converting all customer requirements into technical view.
Tran Nguyen Nhat Linh	Scrum master	Is responsible for promoting and supporting Scrum as defined in the Scrum Guide by helping everyone understand the theory, practices, rules, and values of Scrum...; but also by helping people outside of the Scrum

		Team evaluate the importance of each interaction with them, to understand which of their interactions with the team are useful and which are not.
Doan Thuy Trang	Dev team	Is responsible for transforming the expressed needs into usable functionalities.
Ngo Thanh Van	Dev team	Is responsible for transforming the expressed needs into usable functionalities.
Ho Khai Yen	Dev team	Is responsible for transforming the expressed needs into usable functionalities.

Project Constraints:

Table 2.7. Project Constraints (Source: Authors)

Constraints	Description
Timeline	The project needs to be completed within 2 months.
Budget	The budget should not exceed 6,000 USD for the entire project.
Resources	Need to use a website development team with experience and ability to work with CRM systems.
Policy	Need to comply with data security regulations and GDPR (General Data Protection Regulation)

Cost-Benefits Analysis:

Table 2.8. Cost-Benefits Analysis (Source: Authors)

Cost	Benefit
Website development: \$2,000	Increase revenue from online booking services.
Marketing and advertising costs: \$500	20% revenue growth in 1 year after website launch.

CRM system integration: \$400	Improve customer interaction and optimize marketing strategy.
Annual maintenance and technical support: \$500	Increase customer satisfaction and retention.
Total Cost: \$3,400	Expected ROI: 150% (Estimated revenue growth of 150% after 1 year of website operation.)

2.5. Systems Requirements Documents

2.5.1. Functional Requirements

Functional requirements are mandatory criteria that a system must meet and are verified through various types of testing. They are divided into two main types:

- **Process-Oriented Requirements (PO):** Describes processes that the system needs to perform.
- **Information-Oriented Requirements (IO):** Information that the system needs to store.

To facilitate management and traceability, each requirement is assigned a unique identifier (PUI). For PO, the PUI starts with "PO," followed by the requirement type code and function code (AD: Admin, A: Authentication), ending with a sequential number.

Table 2.9. Functional requirements (Source: Authors)

Req. No.	Title	Description
POA001	Log in validate	When users log in, the system shall validate user credentials.
POA002	Register validates	When users register, the system shall validate user credentials.
POA003	Tokens validate	When users log in, the system shall validate AccessToken.

POAD001	User management	Add, update, view and delete user account
POAD002	Room management	Add, update, view and delete room information
POAD003	Booking management	Update and view booking information
POAD004	Restaurant management	Add, update, view and delete restaurant information
IO01	User Information Storage	The system must contain the information of all users that register to the system
IO02	Room Information Storage	The system must contain the information of all rooms
IO03	Booking Information Storage	The system must contain the information of all booking
IO04	Restaurant Information Storage	The system must contain the information of all restaurants

2.5.2. Non-Functional Requirements

According to BABOK version 3, a Non-functional requirement does not relate directly to the behavior or functionality of the solution, but rather describes the conditions under which a solution must remain effective or the qualities that a solution must have. In other words, Non-Functional Requirements pertain to aspects related to product quality. They encompass various factors such as performance, reliability, and usability, etc. Non-functional requirements are crucial in ensuring that software operates effectively and provides good user experience.

In this project, Non-Functional Requirements include the following categories: Performance Requirements (PR), Platform Constraints (PC), Accuracy and Precision (AP), Adaptability (AD), Security (SC).

Table 2.10. Non-functional requirements (Source: Authors)

Req. No.	Title	Description
PR01	Environments	The hotel reservation system operates on a web platform so customers can easily access it.
PR02	Response Time	The system must respond to user operations within 2 seconds, and process transactions in less than 5 seconds to ensure user experience.
PR03	Update frequency	The system must be updated at least once a month to ensure features and security still work well. Each update should take no more than 30 minutes.
PR04	Capacity	The system is capable of storing and managing reservation data for at least 1,000 customers, 50 rooms and 1,000 transactions.
PC01	Supported Platforms	The system supports Windows and IOS platforms to reach all customers.
PC02	Backend tools	Use MongoDB to store data about users, rooms, restaurants, bookings.
PC03	System Integration	Must be able to integrate with online payment platforms (like PayPal).
AP01	User Identification	Distinguish different users based on username and email. There cannot be 2 users with the same username or email.
AP02	Input Sensitivity	Input data is case sensitive. For example: Username123 is different from username123.
AP03	Admin Notifications	The system will send a notification to the admin if the customer enters the wrong password 5 times, and the transaction fails.
AD01	User Management	Can add/delete/modify users in the admin interface.
AD02	Platform Updates	The system shall adapt to platform updates seamlessly, ensuring no disruption in service for users.

AD03	Admin Access Levels	IT administrators must have role-based access controls to manage user accounts (customer and admin) to perform maintenance tasks without affecting overall system security.
SC01	Login Requirements	To log in, users need to enter the correct password.
SC02	Data Security	The user's password must be hashed using MD5.
SC03	Brute Force Protection	If the user enters the wrong password 5 times, the system will deactivate for 30 minutes.
SC04	Password Recovery	When the user forgets the password, a link to create a new password must be sent to the email address used to register the account.

CHAPTER 3: DATA AND PROCESS MODELING

3.1. Overview and purposes

Chapter 3 plays a crucial role in the analysis and design of information systems by focusing on data and process modeling. This is a key step in the system development process, helping analysts and designers gain a deeper understanding of the structure and operation of the system being built. The chapter comprehensively introduces the main techniques and tools for describing in detail the data structures, information flows, and complex business processes within an organization.

The main purpose of the chapter is to provide readers with a diverse and powerful set of tools for system modeling. The introduction includes widely used modeling methods such as BPMN (Business Process Modeling Notation) and DFD (Data Flow Diagram). BPMN is used to visually describe business processes, while DFD focuses on illustrating data flow within the system.

Another important aspect is explaining how to use context diagrams and different levels of DFD to analyze the system at various levels of detail. The method allows starting with an overview of the system through the context diagram, then gradually delving into more specific details through different levels of DFD. This approach ensures that no aspect of the system is overlooked, and all components are thoroughly analyzed.

One of the important objectives is to support the identification of system requirements through the analysis of data and process models. By creating detailed models of how the system operates, analysts can identify specific requirements that the system needs to meet. The process includes requirements for functionality, performance, security, and scalability of the system.

The content lays a solid foundation for the subsequent stages of the system development process. The models and analyses created will directly support database design and software development.

Chapter 3 not only provides theoretical knowledge about data and process modeling but also equips readers with practical skills to apply this knowledge to real projects. Through understanding and applying the techniques presented, system analysts and designers will be able to create effective information systems that meet the complex needs of modern organizations.

3.2. Deliverables

The deliverables for this chapter encompass several key tasks essential for effective data and process modeling. These tasks include:

- Business Process Modeling (BPMN): Develop a BPMN diagram that visually represents business processes, including tasks, events, gateways, and the flow of activities.
- Requirement Modeling by DFD (Data Flow Diagram): Create a series of DFDs that illustrate how data moves through the system, including:
 - + Data Sources, Data Flows, Data Stores, Data Destinations, and Processes: Identify and document all relevant components that interact within the system.
 - + Context Diagram – Context Level: Construct a context diagram that provides an overview of the system as a single process with its interactions with external entities.
 - + Level 0 of DFD: Develop a Level 0 DFD that breaks down the overall process into its major subprocesses while maintaining high-level abstraction.
 - + Level 1 of DFD: Create a Level 1 DFD that further decomposes the subprocesses identified in Level 0 into more detailed components.
- Data Dictionary: Compile a comprehensive data dictionary that defines all data elements used in the DFDs, ensuring clarity and consistency in terminology.

3.3. Business process modeling and notation (BPMN)

3.3.1. Process overview

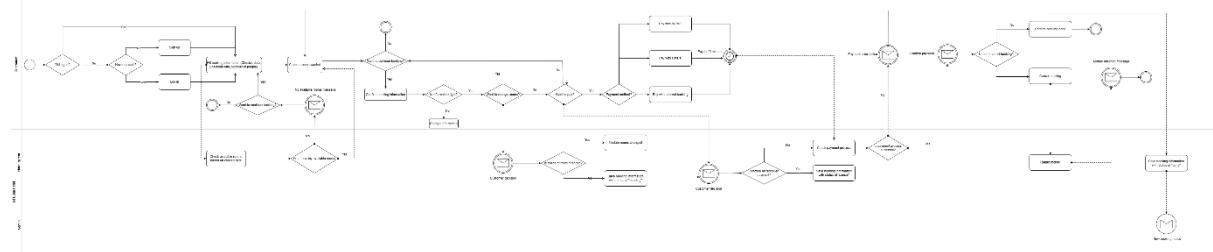


Figure 3. 1. Business process modeling and notation - Process overview (Source: Authors)

Customers need to log in to proceed with hotel booking. The login information includes a username and password. If the customer does not have an account, they can create a new account by providing the following details: Full name, username, email address, and password. Additionally, after logging in, customers can edit their information in the "Edit Information" section and provide other necessary details such as gender and date of birth. These details are not mandatory during account creation, but they are important for the booking process and ensure that the hotel services are appropriate and optimal for the customer.

For the booking process, once customers have registered or logged in, they need to enter booking information, including Check-in date, Check-out date, and the number of people. The system will then display a list of available rooms based on this information. If no rooms are available during the selected dates, the system will suggest the customer choose alternative dates. If rooms are available, customers can view the different room types and select the one that suits them. At this point, if no room is suitable, the customer can change the date to view other options. Once a suitable room is selected, the system will confirm the customer's booking details; if the information is correct, the system will proceed to the payment step. If incorrect, the customer will need to correct the information. Additionally, customers can adjust their room selection after reviewing the booking details, such as removing a room or selecting a different room. If they choose

a different room, the system will return to the available room list for the customer to repeat the selection process.

During the payment step, customers can choose from various payment methods, including MOMO, PayPal, or bank transfer. At this stage, customers can also provide a discount code. If the discount code is valid, the system will adjust the invoice accordingly. If the code is incorrect, the system will display the message "The discount code is invalid or has expired." If the payment is successful, the system will collect the information and save it in the database. If the payment fails, the system will return to the payment step and prompt the customer to repeat the action.

3.3.2. Login/Sign up process

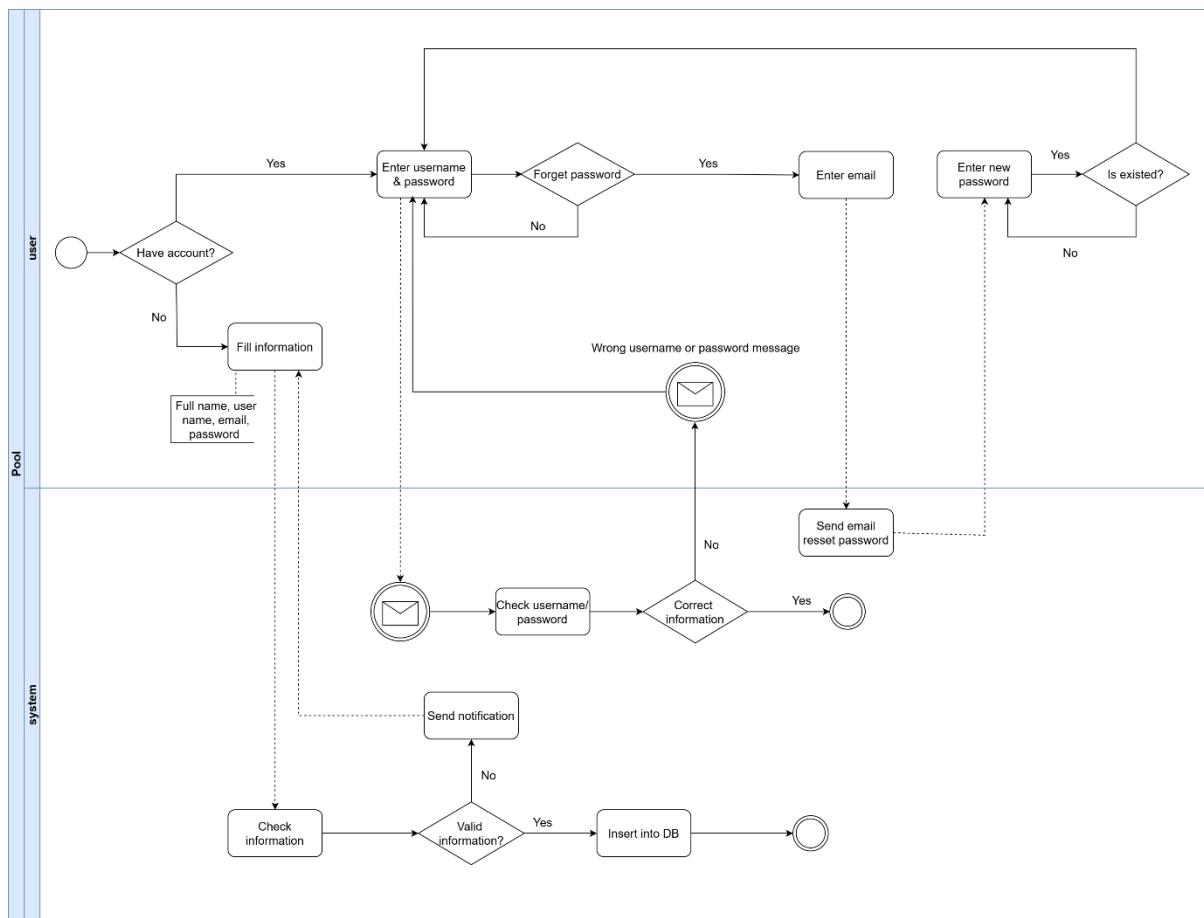


Figure 3. 2. Business process modeling and notation - Login/ Sign up process

(Source: Authors)

For users who already have an account, they can log in to the website by entering their username and password. The system will check their password and username. If incorrect, the user will be asked to re-enter their login information. In case the user forgets their password, they can select the Forgot Password button on the screen, the system will send an account verification link via their email and the user will proceed to set a new password and continue the login process.

For users who do not have an account, they proceed with the account creation process. The user proceeds to provide necessary information such as: Full name, username, email, password. After entering, they will confirm the account information. If the information is incorrect, they must re-enter it; if correct, the system will save the information into the database.

3.3.3. Update the booking (admin)

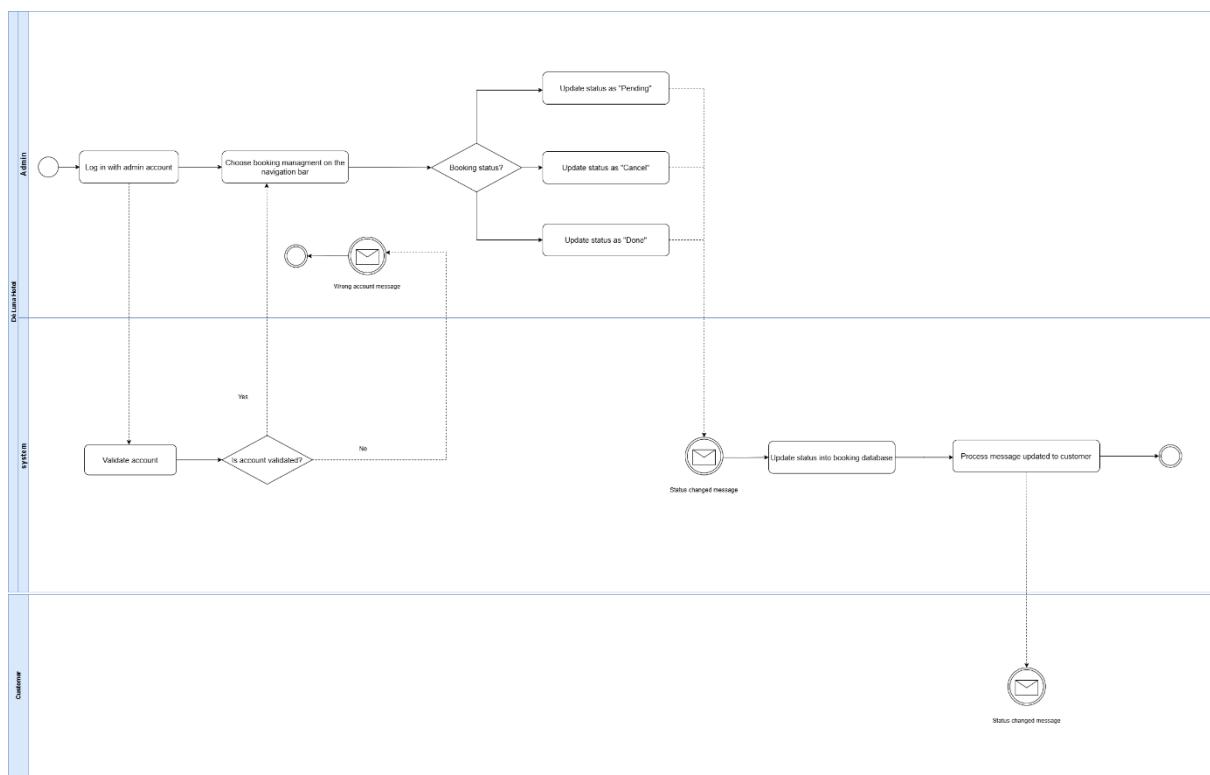


Figure 3. 3. Business process modeling and notation - Update the booking
(Source: Authors)

The process begins with logging into the admin account, after which the administrator can create and manage tracking actions through the management bar. The

system checks the current status of the tracking and updates it to statuses such as "Pending," "Canceled," or "Done" depending on the situation. If the account is invalid, the system will perform account validation. Once the information is valid, the system prepares the message to be sent and updates the status. After updating the status in the database, the system sends a notification to the customer about the new status. This process is designed for managing and tracking statuses, updating information to customers, and can be applied in customer service systems or order tracking systems.

3.4. Requirement modeling by DFD (Data Flow Diagram)

3.4.1. Data sources, data flows, data store, data destinations and process

The data source in the hotel reservation system primarily involves customers and administrative staff. Customers provide essential information, such as personal and account details (name, email, phone number, gender, username, and password) when creating an account or booking a room. Additionally, they input booking specifics, including check-in and check-out dates, room type, quantity, and payment method. On the administrative side, admins manage room details, such as updating room availability, pricing, and booking statuses, as well as providing room descriptions and managing customer information.

Data flows begin when customers interact with the system. They submit account details for registration or login, initiate searches for available rooms based on specific dates and preferences and provide booking details and payment information to complete reservations. After processing, the system returns search results, booking confirmations, or notifications. Payment information is sent from the system to an external payment gateway, which then returns confirmation or failure statuses. Admins also send updates to the system, managing room availability, prices, and booking status changes as required.

Data stores play a crucial role in managing various customers and booking details. The Users Data Store keeps all customer information, including personal data, booking history, and account credentials, ensuring secure authentication and access control. The Rooms Data Store holds information about room types, availability, pricing, and

descriptions, facilitating the search and booking processes. The Bookings Data Store records all booking details, such as check-in and check-out dates, room type, total cost, and booking status (e.g., pending, cancelled, confirmed), ensuring smooth reservation management.

Data destinations support the operational workflow. The Customer receives booking confirmations, cancellations, and notifications related to their account. The Admin accesses updates on room availability, booking statuses, and can generate reports for operational insights. The Payment Gateway handles payment requests and responds with transaction statuses, which are then relayed to the customer and stored in the system.

Key processes within the hotel reservation system encompass a range of customer and administrative activities. During registration or login, customers and admins enter account details to access the system. The Browse & Search process allows customers to search for available rooms based on specific criteria, accessing the Rooms Data Store to retrieve results. Admins manage room details in the Rooms Data Store by adding, updating, or deleting information on room availability, types, and pricing. The Book Room process involves verifying room availability and updating the Bookings Data Store with new reservations. The Apply Payment process collects payment details, sends them to the payment gateway for processing, and updates the transaction status accordingly. Admins can update booking statuses, confirming check-ins or cancellations, which are reflected in the Bookings Data Store. Finally, the system sends confirmation emails or notifications with reservation and invoice details to customers, ensuring they receive all necessary booking information after a successful transaction.

3.4.2. Context diagram – Context level

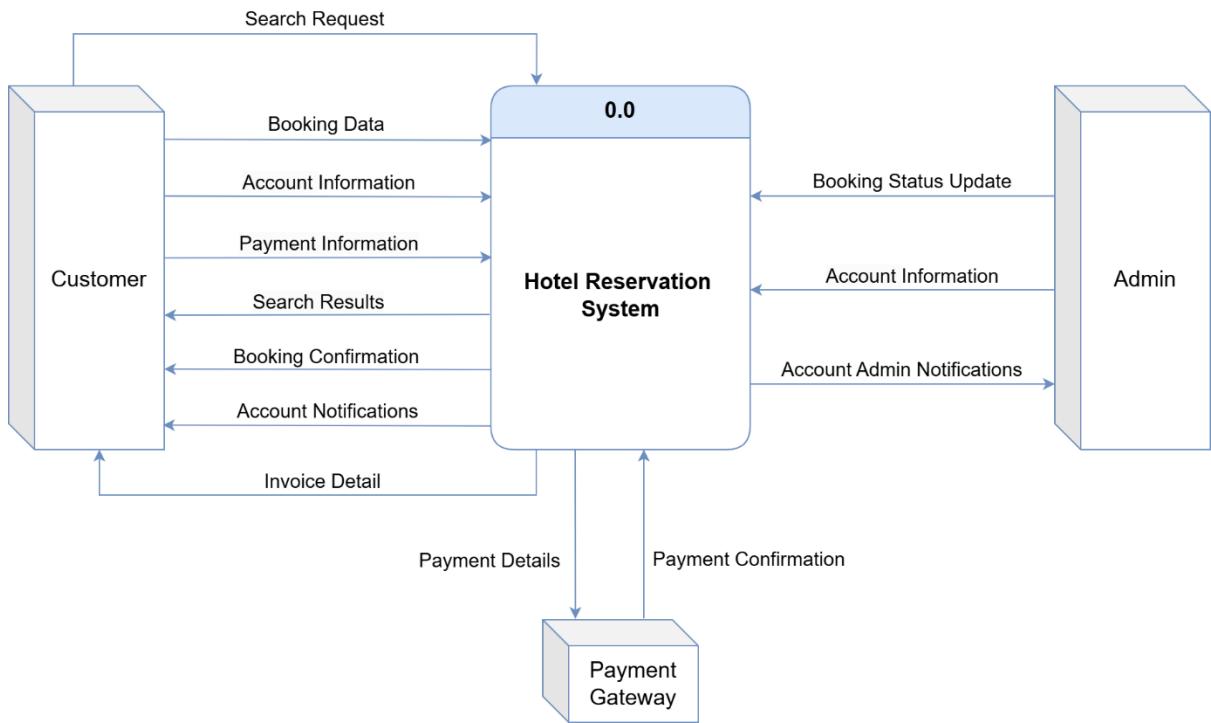


Figure 3. 4. Context Diagram (Source: Authors)

The Context Level Data Flow Diagram (DFD) provides a detailed overview of the hotel booking system's interactions with external entities, capturing the structured flow of information between various operational components. Central to this diagram is the Hotel System, which serves as the primary process node, interfacing with multiple stakeholders through dedicated data flows. The system's architecture illustrates a complex network of data exchanges, with distinct bidirectional communication flows centered on the Customer entity. This interface involves receiving input, such as search criteria, booking data, and payment details, and delivering outputs like search results, booking confirmations, invoices, and account notifications. The Customer entity is unique in its bidirectional communication pattern, while interactions with the Admin and Payment Gateway occur in a predominantly unidirectional manner.

The diagram further depicts the integration of internal operational units, with data flows directed toward maintaining efficient room management and booking processes. The Admin manages Booking Status Updates, adjusting booking statuses as necessary, while their login process involves a secure channel for Admin Account Information.

Admin accounts are pre-set, eliminating the need for registration or account creation, although they can access customer account data as needed.

A key component of the system is its Payment Gateway integration, which processes financial transactions when the customer submits payment details. Upon successful processing, the gateway communicates a Payment Confirmation, or in cases of failure, a rejection notice, both of which update the booking status within the system.

This context-level representation effectively establishes the boundaries and interactions within the hotel booking system, demonstrating how data flows support efficient hotel operations and facilitate seamless communication between the system and its stakeholders. The architecture underscores the significance of structured data exchanges, which enable reliable information management and operational agility in responding to customer needs and administrative updates.

3.4.3. Level 0 of DFD

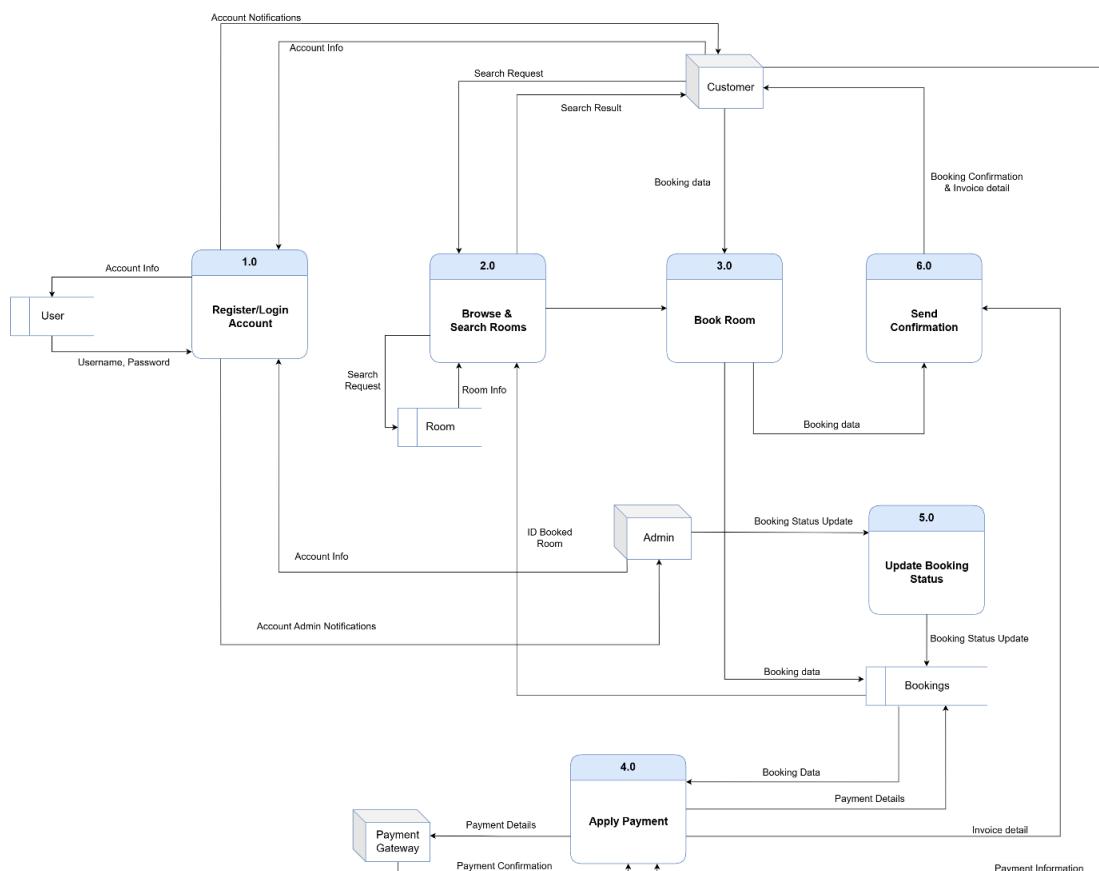


Figure 3. 5. Level 0 of DFD (Source: Authors)

This Level 0 Data Flow Diagram provides a comprehensive breakdown of the operational processes within a hotel booking system, highlighting an interconnected network of essential functions and data flows. The diagram encompasses six key processes, numbered from 1.0 to 6.0, each representing a critical phase in the customer journey and the overall management of hotel operations.

The system begins with Process 1.0 (Register/Login Account), where customers initiate their interaction by creating an account using personal information such as full name, contact details, username, and password. Admins, however, bypass this step with pre-set accounts and directly log in. Upon successful registration or login, the system notifies users with account-related updates, ensuring readiness for subsequent actions.

Next, Process 2.0 (Browse & Search Rooms) enables customers to explore available accommodations. This process allows for room browsing without requiring login credentials. Customers can refine their search using criteria such as check-in and check-out dates, room type, and the number of occupants. The system retrieves matching room data by querying both the Rooms Data Store to get list room and Booking Data Store to get ID booked room, ensuring up-to-date availability information.

The workflow advances to Process 3.0 (Book Room), where customers finalize their reservation after selecting a preferred room from the search results. This process requires prior completion of the login and search phases. By clicking the "book" button, customers initiate the reservation, prompting the system to update booking records within the Booking Data Store.

The critical financial step occurs at Process 4.0 (Apply Payment), which collects payment information and forwards it to an integrated external Payment Gateway. The gateway processes the transaction and returns a status—either confirmation or failure—allowing the system to update the payment status accordingly.

For backend operations, Process 5.0 (Update Booking Status) empowers admins to manage booking details, including confirming check-ins or canceling reservations. These updates reflect directly in the Booking Data Store, maintaining accurate and real-time booking statuses such as "pending," "cancelled," or "done."

The final phase, Process 6.0 (Send Confirmation), completes the booking journey by generating a detailed confirmation for the customer. This confirmation, sent via email or in-app notification, includes essential information such as reservation details and invoice data, providing customers with documentation and assurance for their stay.

3.4.4. Level 1 of DFD

3.4.4.1. Process 1: Register/Login Account

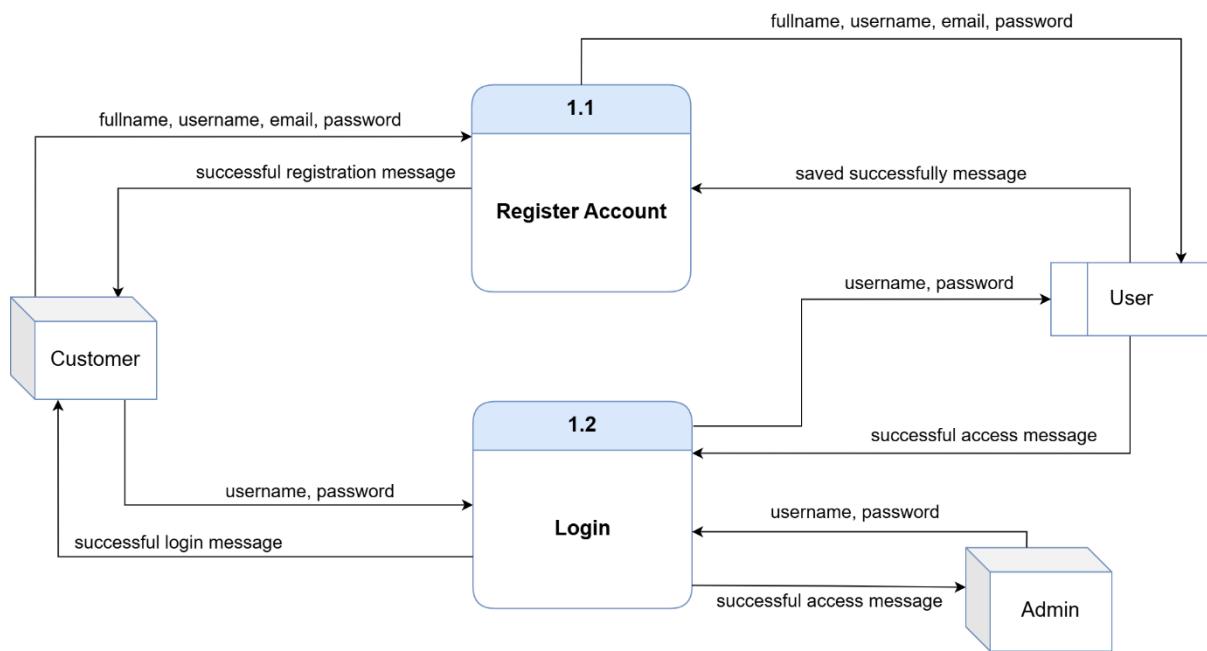


Figure 3. 6. DFD level 1 of process 1 (Source: Authors)

This Level 1 Data Flow Diagram provides a detailed breakdown of the user account management processes within a hotel booking system, highlighting the critical steps in registration and login functionality. The diagram features two main processes, numbered **1.1** and **1.2**, representing the stages from account creation to system access, which are essential for both customer interactions and administrative control.

The process begins with **Process 1.1 (Register Account)**, where new customers initiate account creation by providing essential personal information, including their **full name, email, username and password**. The system then processes this data, validating it for accuracy and storing it within the **User data store**. Upon successful registration, the system sends a **successfully registration message** back to the

customer, confirming that their account has been set up and saved in the system. This process ensures that each customer has a unique and verified account, allowing secure and personalized access.

Following registration, **Process 1.2 (Login)** allows both customers and admins to enter the system using their credentials (**username and password**), which the system verifies against the stored data in the **User data store**. If authentication is successful, a **successful access message** is returned. This step facilitates **role-based access control**, distinguishing between customers who are booking rooms and admins who oversee system operations. Once a user's credentials are verified, the system grants **access rights** based on the user's role, allowing them to proceed to their respective interfaces **customer dashboard** or **admin dashboard**. The login process is essential for maintaining security and ensuring that only authenticated users access the system.

3.4.4.2. Process 2: Browse & Search Rooms

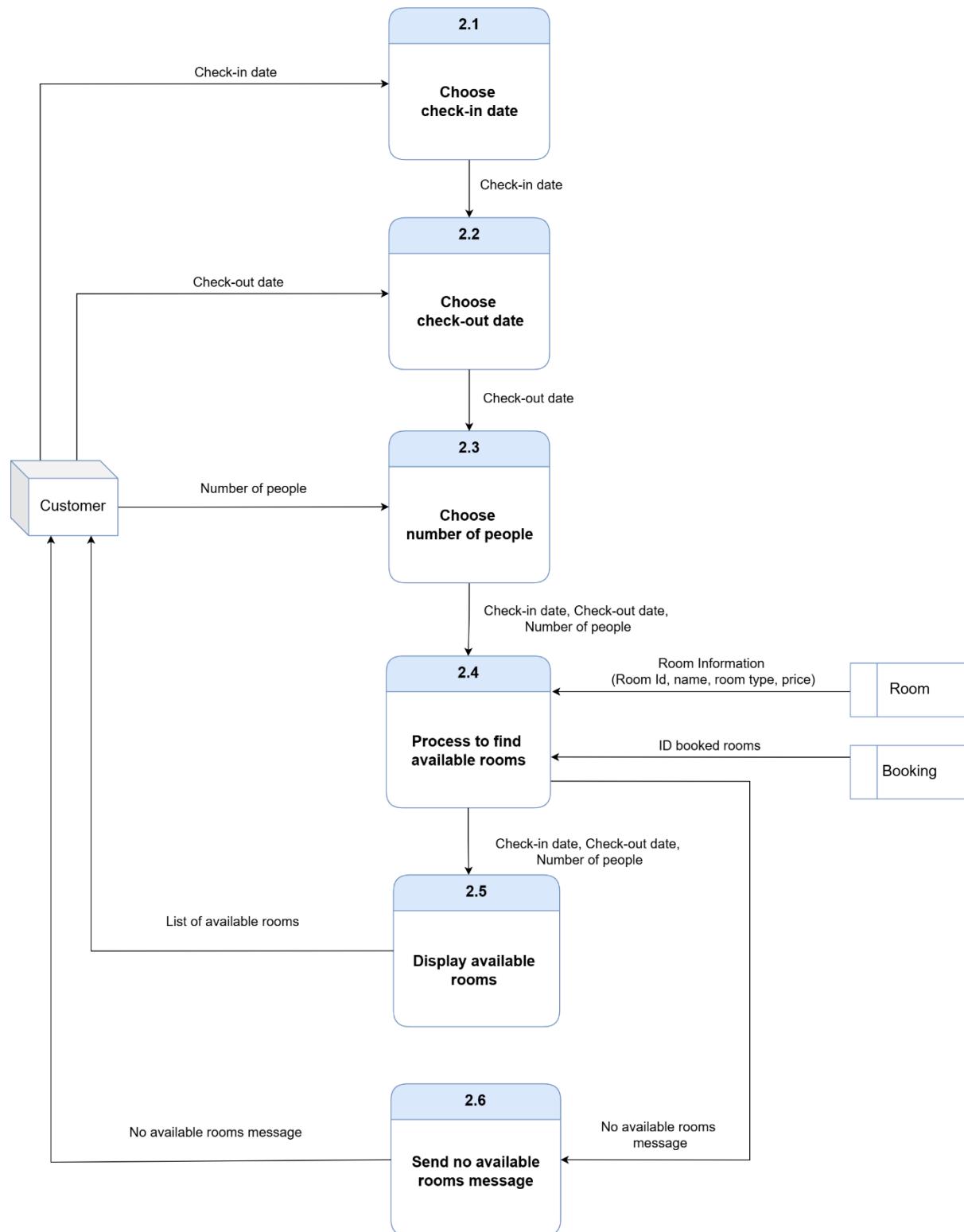


Figure 3. 7. DFD level 1 of process 2 (Source: Authors)

This Level 1 Data Flow Diagram provides a detailed breakdown of the room browsing and search processes within a hotel booking system, outlining the steps that guide customers in selecting and viewing available rooms based on their specific requirements. The diagram features six main processes, numbered **2.1** through **2.6**, which together facilitate an organized flow from choosing check-in and check-out dates to displaying room availability or notifying customers when no rooms are available.

The process begins with **Process 2.1 (Choose Check-in Date)**, where customers specify their intended check-in date. This date information is sent to the system, which processes it to ensure it aligns with the available booking schedule.

Next, in **Process 2.2 (Choose Check-out Date)**, customers enter their check-out date. This date is essential for determining the room's availability over the customer's desired stay period, prompting the system to check for open room slots that match the selected check-in and check-out range.

Following this, **Process 2.3 (Choose Number of People)** allows customers to specify the number of people who will be staying. The system uses this information to filter room options based on capacity, ensuring that only rooms capable of accommodating the specified number of guests are considered.

Once the check-in date, check-out date, and number of people have been submitted, **Process 2.4 (Process to Find Available Rooms)** queries the **Room Data Store** to get list room and the **Booking Data Store** to get ID booked room. It retrieves information about available rooms by considering room details (such as room ID, name, type, and price) and excludes rooms that are already booked within the requested time frame.

Based on the query results, two scenarios are possible: In **Process 2.5 (Display Available Rooms)**, the system displays a list of available rooms that match the customer's criteria, including essential details about each room to aid in the booking decision. Alternatively, if no rooms meet the requirements, **Process 2.6 (Send No Available Rooms Message)** notifies the customer with a message indicating that no rooms are available for the specified dates and conditions.

3.4.4.3. Process 3: Book Room

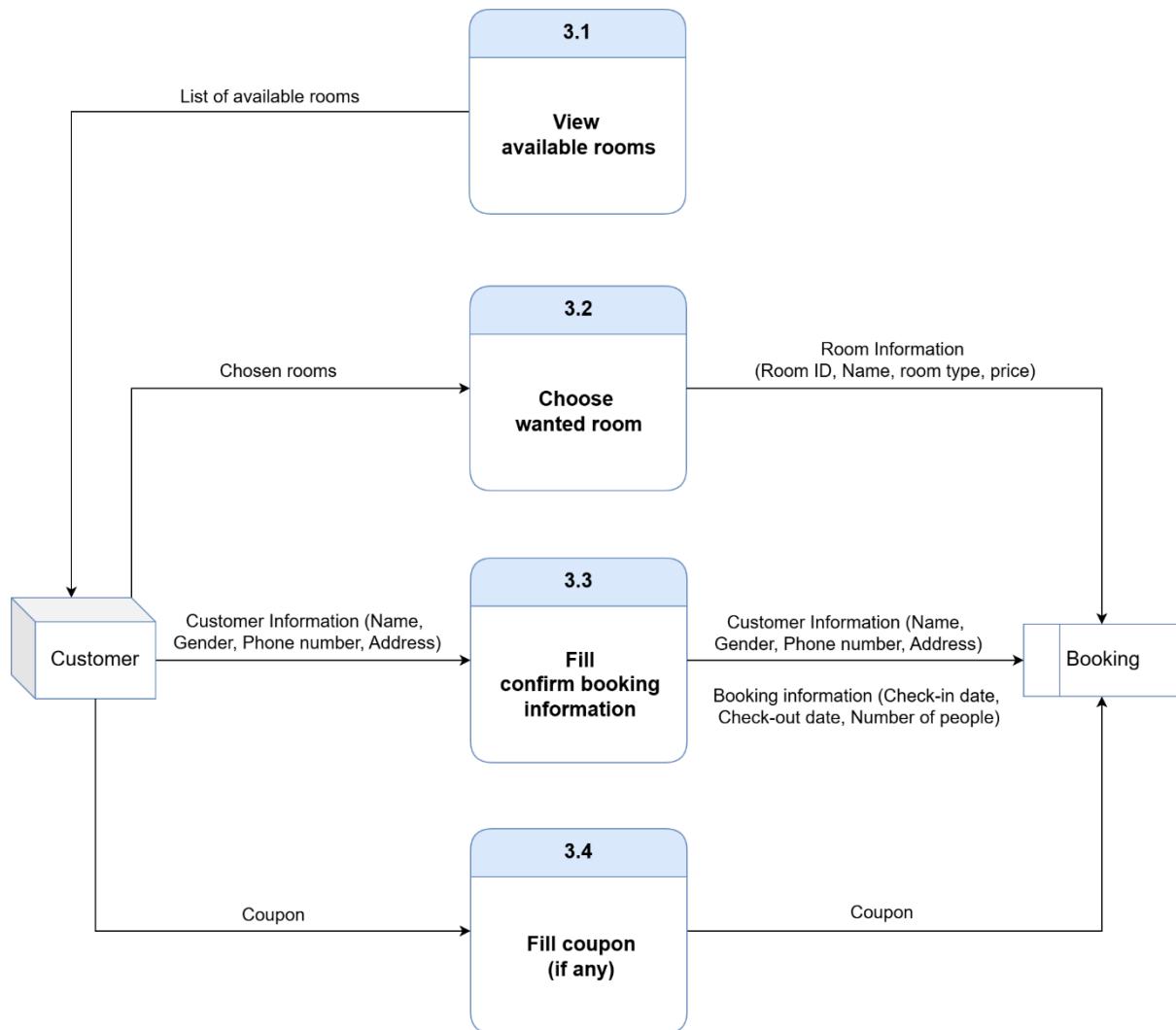


Figure 3. 8. DFD level 1 of process 3 (Source: Authors)

This Level 1 Data Flow Diagram (DFD) details the room booking process in the hotel booking system, outlining each step the customer takes to reserve a room, from viewing available rooms to finalizing the booking with optional coupon entry. The diagram illustrates four main processes (3.1 to 3.4), ensuring a seamless reservation experience.

The process begins with **Process 3.1 (View Available Rooms)**, where customers access a list of available rooms filtered by criteria such as check-in/out dates and the number of guests. This step presents key details (e.g., Room ID, Name, Room Type, Price) to facilitate informed decision-making.

In **Process 3.2 (Choose Wanted Room)**, customers select their preferred room. The system retrieves detailed information on the selected room, confirming its availability and specifications to proceed with the booking.

Following room selection, **Process 3.3 (Fill Confirm Booking Information)** requires the customer to provide personal details (Name, Gender, Phone Number, Address) and booking specifics (Check-in Date, Check-out Date, Number of Guests). These inputs ensure the booking is correctly linked to the customer and logged in the Booking repository.

Lastly, **Process 3.4 (Fill Coupon, If Any)** allows customers to apply a discount or promotional coupon. The system validates the coupon and applies any eligible discount to the booking, updating the total cost accordingly.

3.5. Data Dictionary

3.5.1. Data flow

The data in the hotel booking system flows through several steps, starting from when users register their account to when they receive the booking confirmation and detailed invoice. First, users provide account information such as full name, email, username, and password, which is stored and used for the account registration process. Next, users perform a room search by providing details such as check-in date, check-out date, and the number of guests. The system retrieves room information, including room name, price, and images, from the data store and displays the search results for the user.

When users decide to book a room, booking information, including user ID, room ID, check-in date, check-out date, special requests, etc., are sent to the system for processing. This data is stored and later used for the payment process. The system calculates the total amount and requests payment, sending the information to the payment gateway. Once the payment is completed, payment confirmation information, including payment status and booking ID, is updated in the system.

Finally, after a successful payment, the system sends a booking confirmation and invoice details to the user, including booking ID, contact information, total amount, and

payment status. This process ensures a seamless booking experience and creates a continuous data flow from the user's room search, booking, and payment to the receipt of confirmation.

No.	Name	Description	Origin	Destination	Record	Volume and Frequency
1	Account information	Data related to the user, including fullname, email, username and password	Customer Entity	Register Account Process	fullname	High volume, updated frequently when user register
2	Search Request	Data for room search, including keywords, check-in date, check-out date, and number of guests			email	
3	Room Info	Room information, including name, price and images			username	
4	Search Result	Room information, including name, price and images			password	
5	Booking data	Data of booking information, including customer and room details	Customer Entity	Browse & Search Rooms Process	check_in_date	High volume, existed when the user searches for rooms
6	Booking data	Data of booking information, including customer contact information, room details and customer partner information			check_out_date	
7	Booking data	Data of booking information to apply payment process	Book Room Process	Bookings Data Store	room_guest	High volume, existed when the user request made a booking
8	Payment Details	Information required for payment processing.	Book Room Process		room_id	
9	Payment Confirmation	Information for confirmation payment .	Book Room Process	Apply Payment Process	check_in_date	High volume, existed when the user request made a booking
10	Payment Details	Information after payment confirmation	Book Room Process		check_out_date	
11	Invoice Details	Finalized invoice details for the booking.	Book Room Process	Send Confirmation	contact_infor	Low volume, sent after booking confirmation
12	Booking Status Update	Updates the booking status after payment	Book Room Process		checkin_time	
13	Account notifications (customer)	Notifies message successful or not successful	Send Confirmation	Customer Entity	total_amount	Medium, existed when the user request make a payment
14	Account notifications (admin)	Notifies message successful or not successful	Customer Entity		special_request	
15	Booking Confirmation & Invoice detail	Final confirmation of the booking and associated invoice details.	Customer Entity		user_id	Medium volume, processed for each payment transaction
					room_id	
					check_in_date	Low volume, sent after booking confirmation
					check_out_date	
					payment status	
					total_amount	
					user_id	
					contact_infor	
					room_id	

Figure 3. 9. Data flow (Source: Authors)

3.5.2. Data store

The User Data Store stores essential information about both users and admins, including personal details and authentication credentials. Key attributes include user_id, username, password, fullname, email, phone_number, avatar, gender, dob, and address. It also manages account status and verification information, such as status, verified, isAdmin, resetPasswordToken, and emailVerificationToken, alongside important timestamps like createdAt and updatedAt to track account activity. The Room Data Store holds comprehensive information about each hotel room, such as room_id, room_number, room_name, room_price, and room_type, as well as its current availability (room_status). Detailed room features are stored under room_details, which includes information like bed types, occupancy capacity, size, bathroom features, view, and unique attributes. Additionally, it stores amenities (room_amenities), images (room_img), and timestamps (createdAt, updatedAt) for room management. The Bookings Data Store store all booking details, including booking_id, check_in_date, check_out_date, and customer contact information (e.g., full_name, gender, phone_number, email, and address). It also includes guest information (guest_id, guest.name), booking-specific details such as checkin_time, special_request, booking_status, and total_amount. Each booking is associated with a specific room (room_id) and user (user_id), and is time-stamped with createdAt and updatedAt to ensure accurate tracking.

These data stores work together to efficiently manage and retrieve user, room, and booking data, ensuring smooth operation throughout the hotel booking system.

No.	Name	Description	Attribute
1	User	Store information about user and admin, including credential and personal information	user_id username password fullname email phone_number avatar gender dob address status verified isAdmin resetPasswordToken emailVerificationToken emailVerificationExpire createdAt updatedAt
2	Room	Store information about room, including room details, room amenities,,etc	room_id room_number room_name room_price room_type room_status room_details_id room_details.beds room_details.occupancy room_details.size room_details.bathroom room_details.view room_details.unique_feature room_amenities room_img createdAt updatedAt
3	Bookings	Holds data booking , including details for each booking	booking_id check_in_date check_out_date contact_infor.full_name contact_infor.gender contact_infor.phone_number contact_infor.email contact_infor.address guest_id guest.name checkin_time special_request booking_status total_amount room_id user_id createdAt updatedAt

Figure 3. 10. Data store (Source: Authors)

3.5.3. Entity

No.	Name	Description	Input data flow	Output data flow
1	Customer	Represents the customers interacting with the system to perform operations such as account registration, room search, booking, and receiving notifications.	Account Information	Account Notifications
			Search Request	Search Result
			Booking Data	Booking Confirmation & Invoice Detail
2	Admin	Represents administrators managing bookings.	Account Information	Account Notifications
				Booking Status Update
3	Payment Gateway	External payment processing system.	Payment Details	Payment Confirmation

Figure 3. 11. Entity (Source: Authors)

3.5.4. Process

The system follows a structured process that begins with the **Register/Login** process, where users can either create a new account or log into an existing one by providing account information, resulting in account notifications such as registration confirmation or login status updates. Next, users can **Browse and Search Rooms**, where they enter search criteria (e.g., room type, check-in/check-out dates, number of guests) to receive room information and a list of available rooms that match the search criteria.

After selecting a room, users proceed to **Book a Room**, where they provide booking details such as room selection, dates, and guest information. This data is then stored in the **Bookings Data Store** for further processing. Following the booking, the **Payment** process allows users to complete the payment by providing payment details, and payment information is processed to confirm the transaction, returning a payment confirmation along with payment details.

Once the payment is confirmed, the **Update Booking Status** process is triggered, allowing the admin to update the booking status (e.g., "Confirmed", "Paid") based on the successful transaction. Finally, the **Send Confirmation** process ensures that the customer receives an email confirmation with booking details and an invoice, marking the completion of the booking process. This entire workflow ensures a seamless and efficient experience for both users and the system.

No.	Name	Description	Input data flow	Output data flow
1	Register/Login	Process to register a new user account or login to an existing account.	Account Information	Account Notifications
2	Browse & Search Rooms	Process to browse and search for available rooms.	Search Request	Room Info, Search Result
3	Book Room	Process to allow customers to make a booking for a selected room	Booking data (from Customer)	Booking data (to Bookings Data Store)
4	Apply Payment	Process to handle the payment for a booking.	Booking data (from Bookings Data Store), Payment Details	Payment Confirmation, Payment Details
5	Update Booking Status	Process to update the status of booking details after successful payment.	Booking Status Update (from Admin)	Booking Status Update
6	Send Confirmation	Process to send booking information and billing invoices to customers.	Booking data (from Bookings Data Store), Invoice Details	Booking Confirmation & Invoice detail (to Customer)

Figure 3. 12. Process (Source: Authors)

CHAPTER 4: OBJECT MODELING

4.1. Overview and purposes

Chapter 4 focuses on Object Modeling, a critical aspect of software design that emphasizes the representation of systems through various models. The primary purpose of this chapter is to provide a comprehensive understanding of how to analyze and visualize the components and interactions within a system using Unified Modeling Language (UML) diagrams.

The chapter aims to achieve the following objectives:

- Identify Key Components: Recognize and define the essential entities or classes in the problem domain, which are crucial for understanding system requirements.
- Model System Functionality: Utilize different UML diagrams to depict system functionality and interactions effectively, facilitating clearer communication among stakeholders.
- Enhance System Design: Provide tools and methodologies that help in designing robust systems by illustrating relationships, behaviors, and states of objects within the system.

Through these objectives, the chapter seeks to enhance the reader's ability to create accurate models that reflect the real-world scenarios they aim to address, ultimately leading to better software solutions.

4.2. Deliverables

The deliverables for this chapter encompass several key modeling tasks that contribute to a thorough understanding of object-oriented design. These tasks include:

- Scenarios: Develop scenarios that illustrate how users will interact with the system, providing context for the subsequent models.
- Models and Functionality: Create models that represent the functionality of the system, ensuring that all user interactions are captured.

- Use Case Diagram: Construct a use case diagram that visually represents the system's functional requirements by showing actors and their interactions with use cases.
- Use Case Descriptions (User Story): Write detailed use case descriptions or user stories that outline specific functionalities from the user's perspective, including preconditions, main flows, alternative flows, and postconditions.
- Class Diagram: Develop a class diagram that defines the structure of the system by illustrating classes, their attributes, methods, and relationships.
- Sequence Diagram: Create sequence diagrams to depict how objects interact in a particular scenario over time, showcasing message exchanges between objects.
- Activity Diagram: Design activity diagrams to represent workflows or processes within the system, highlighting various activities and decision points.

4.3. Scenarios, Models and Functionality

4.3.1. Use Case Diagram

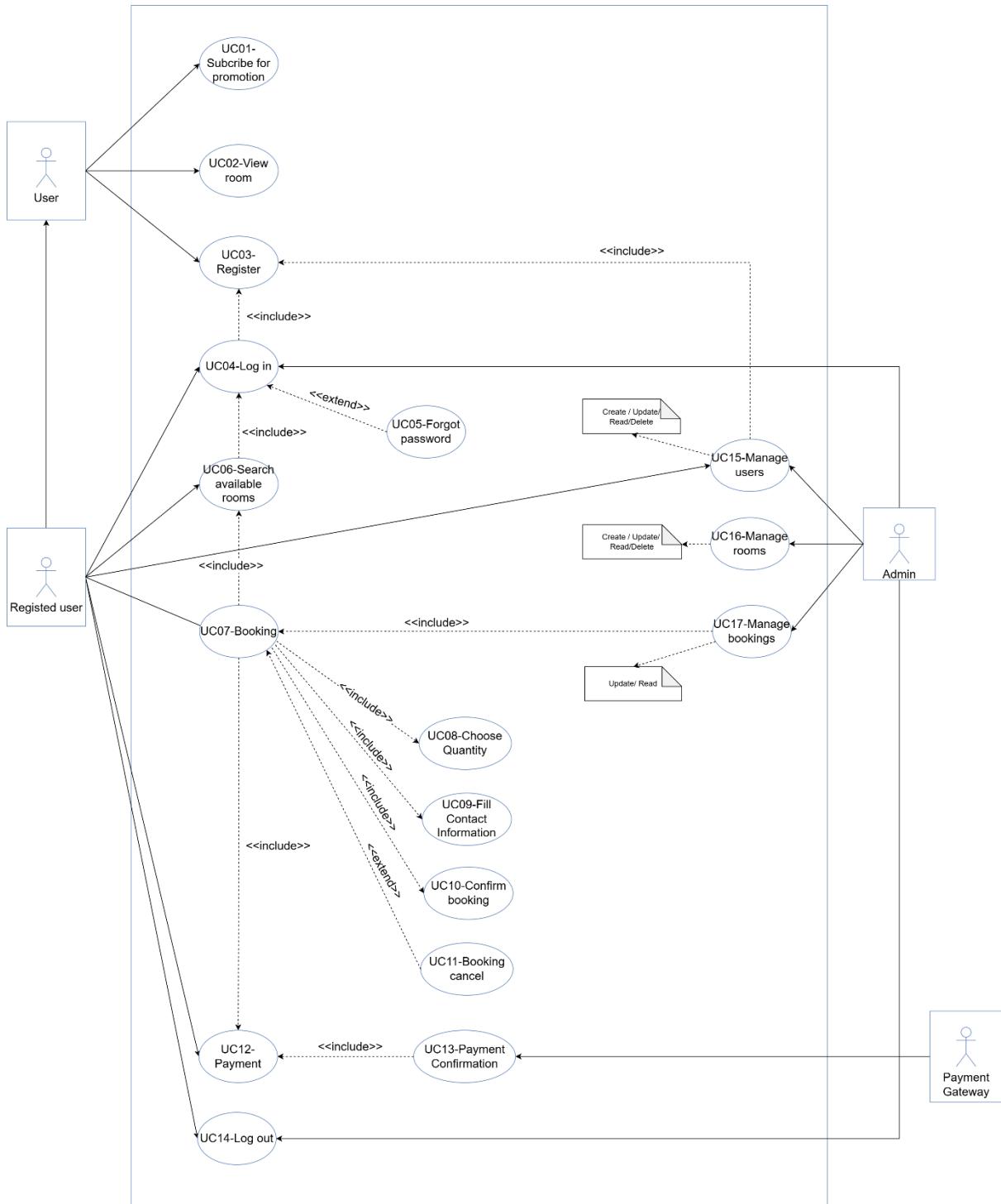


Figure 4. 1. Use Case Diagram (Source: Authors)

4.3.2. Use Case Descriptions (User story)

Table 4. 1. Use case specification of Subscribe for promotion (Source: Authors)

Use Case Name	Subscribe for promotion	Use Case Description
UC #	01	
Primary Actor	User	User subscribes to receive promotional emails from the company. User can enter their email. The system will save it in database email promotion
Use Case Story	User subscribes to receive promotional emails from the company.	
Pre-Condition	User has access to page that have subscribe field (except “Accommodation” and “Service” page)	
Title: User subscribes email for promotion		
Primary Flow (PF)	Action Response	System Response
	1. The user enters their email address into the subscription field.	
	2. The user clicks the “Send me” button.	
		3. The system validates the email format. If valid, go to next step If not valid, go to EF1
		4. The system check the email already in database If not yet, go to next step If already exist, go to EF2
		5. The system navigate user to main screen

		6. The system sends a confirmation message to the user's email
	7. User now receive email promotion on their devices.	
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1) User click any other pages hyperlink button.	
	2) User can return to PF step 1.	
Exception Flow (EF1)	Title: Invalid Email Format	
	Action Response	System Response
	1) System detects that the Email is invalid.	
		2) System displays an error message
Exception Flow (EF2)		3) System return user to PF step 1.
	Title: Email has already existed	
	Action Response	System Response
	1) System detects that the Email has already saved in database	
		2) System displays an error message notifies
		3) System return user to PF step 1.
	The user's email is successfully added to the system's mailing list for promotions.	

Table 4. 2. Use case specification of View Room (Source: Authors)

Use Case Name	View Room	Use Case Description
UC #	02	
Primary Actor	User	Users click specific room to view its detail
Use Case Story	User views the details of a specific room	
Pre-Condition	User selects a specific room to view more details	
Primary Flow (PF)	Title: User view specific room	
	Action Response	System Response
	1. The user selects a room from the list of available rooms.	
		2. The system check selected room If valid, go to next step If invalid, go to EF1
		3. The system displays detailed information about the selected room, including: <ul style="list-style-type: none">- Room Images- Room Type- Amenities- Number of Beds- Capacity If user does not click another page, go to next step If user click another page, go to AF1

	4. The user reviews the room details to decide whether to book it.	
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1. Users click any other page's hyperlink button.	
	2. User can return to PF step 1.	
Exception Flow (EF1)	Title: Selected room is available to view	
	Action Response	System Response
	1. System detects that the room is not able to view	
		2. System displays an error message notifies
		3. System return user to PF step 1.
Post Condition	The user successfully views detailed information about the room	

Table 4. 3. Use case of Register (Source: Authors)

Use Case Name	Register	Use Case Description
UC #	03	
Primary Actor	User	Users register an account in the system. Users enter their full name, email, username, password to create new account
Use Case Story	User register an account in the system.	
Pre-Condition	User has access to the website.	

Title: User register new account to system	
Action Response	System Response
1. Users select “Create Account”	
	2. The system navigate User to “Register Page” If user select another page, go to AF1
3. User fills out the registration form with the following details: <ul style="list-style-type: none">- Full name- Email- Username- Password	
4. User click “Register”	5. System check if all required fields are filled If valid, go to next step If invalid, go to EF1
	6. System check if email already existed If already existed, go to EF2 If not exist, go to next step
	7. System check username already exist If already existed, go to EF3 If not exist, go to next step
	8. System create new account and save in database

		9. System send email verification to user
	10. User now receive verification email and can use new account to log in.	
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1. Users click any other pages hyperlink button.	
	2. User can return to PF step 1.	
Exception Flow (EF1)	Title: Invalid Require Fill	
	Action Response	System Response
	1. System detects that user miss one or some field.	
		2. System displays an error message
Exception Flow (EF2)		3. System return user to PF step 1.
	Title: Email has already existed	
	Action Response	System Response
	1. System detects that the Email has already saved in database	
Exception Flow (EF3)		2. System displays an error message notifies
		3. System return user to PF step 3.
	Title: Username has already existed	
	Action Response	System Response

	1. System detects that the Email has already saved in database	
		2. System displays an error message notifies
		3. System return user to PF step 3.
Post Condition	The user's account is successfully created and stored in the system. The user can now log in with their new account credentials.	

Table 4. 4. Use case specification of Login (Source: Authors)

Use Case Name	Login		Use Case Description
UC #	04		
Primary Actor	Registered user		
Use Case Story	Registered user login to website		Registered user login with their username and password.
Pre-Condition	User already registered		
Primary Flow (PF)	Title: User login		
	Action Response		System Response
	1. Users select “Login” button in header		
			2. The system navigate User to “Login Page” If user select another page, go to AF1
	3. The user enters their username and password.		

	4. Users click “Login” button	
		5. System check if all required fields are filled If valid, go to next step If invalid, go to EF1
		6. System valid username and password If invalid, go to EF2 If valid, go to next step
		7. System navigate user to “Main Page”
	8. User now already log in to website	
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1. Users click any other pages hyperlink button.	
Exception Flow (EF1)	2. User can return to PF step 1.	
	Title: Invalid Require Fill	
	Action Response	System Response
Exception Flow (EF2)	1. System detects that user miss one or some field.	
		2. System displays an error message
		3. System return user to PF step 1.
Exception Flow (EF2)	Title: Email has already existed	
	Action Response	System Response
	1. System detects that the Email	

	has already saved in database	
		2. System displays an error message notifies
		3. System return user to PF step 3.
Exception Flow (EF3)	Title: Username has already existed	
	Action Response	System Response
	1. System detects that the username already exist	
		2. System displays an error message notifies
		3. System return user to PF step 3.
	User login successfully	
Post Condition		

Table 4. 5. Use case specification of Forgot Password (Source: Authors)

Use Case Name	Forgot Password		Use Case Description
UC #	05		
Primary Actor	Registered user		Registered user reset new password
Use Case Story	Registered user reset new password		
Pre-Condition	User already register.		
Primary Flow (PF)	Title: User reset password		
	Action Response	System Response	
	1. Users select “Login” button in header		
		2. The system navigate User to “Login Page” If user select another page, go to AF1	
	3. Users click “Forgot Password” button		

		4. The system navigate User to “Forgot Password Page” If user select another page, go to AF1
	5. Users enter their email address	
		6. System check if email is valid If valid, go to next step If invalid, go to EF1
		7. System sends an email with a link to reset the password.
	8. Users click hyperlink in email	
		9. System navigate User to “Reset Password Page” If user select another page, go to AF1
	10. Users enter New Password and Confirm New Password	
		11. System check if New Password match Confirm New Password If match, go to next step If not match, go to EF2
		12. System save new password in database and navigate user to “Successful Reset Password” page
	13. User now can log in with new password	
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1. Users click any other pages hyperlink button.	
Exception Flow (EF1)	Title: Invalid Email Format	
	Action Response	System Response
	1. System detects that the Email is invalid.	
		2. System displays an error message

		3. System return user to PF step 5.
Exception Flow (EF2)	Title: Password Match	
	Action Response	System Response
	1. System detects that the password not match confirm password	
Post Condition	The user's password is successfully updated in the system. The user can now log in with the new password.	2. System displays an error message notifies
	3. System return user to PF step 10.	

Table 4. 6. Use case specification of Search available room (Source: Authors)

Use Case Name	Search available rooms	Use Case Description
UC #	06	Registered user searches for available rooms based on date
Primary Actor	Registed user	
Use Case Story	Registered user searches for available rooms based on specific criteria	
Pre-Condition	User already login.	
Primary Flow (PF)	Title: User reset password	
	Action Response	System Response
	1. Users select "Room" in header	2. The system navigate User to "Room Page"

		If user select another page, go to AF1
	3. Users choose date “Date Arrival” and “Date Departure” and number of guests	
	4. Users click “Book” button	
		5. System valid Date If valid, go to next step If invalid, go to EF1
		6. System display all available rooms
		7. The system navigate User to “Booking list” page If user select another page, go to AF1
	8. User now can view all available rooms	
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1. Users click any other page's hyperlink button.	
Exception Flow (EF1)	Title: Invalid Date	
	Action Response	System Response
	1. System detects that <ul style="list-style-type: none">- Date Arrival are same with Date Departure- Date Arrival after Date Departure	

	- Date is in the past	
		2. System displays an error message
		3. System return user to PF step 3.
Post Condition	The system displays a list of available rooms based on the specified criteria	

Table 4. 7. Use case specification of Booking (Source: Authors)

Use Case Name	Booking	Use Case Description
UC #	07	
Primary Actor	Registed user	Registered user booking room. Which include search available room (UC06), choose quantity (UC08), Fill contact information (UC09), Confirm Booking (UC10), Payment (UC12)
Use Case Story	Registered user booking room	
Pre-Condition	User already login. User has selected a room The selected room is to be available	
Primary Flow (PF)	Title: User booking rooms	
	Action Response	System Response
	1. User selected activity “Search available room”, the UC06 is performed	
		2. System executes UC06: Displays all available rooms for the selected dates.

	3. Users choose quantity, the <i>UC08</i> is performed	
		4. System executes UC08: Captures the quantity and display in receipt
	5. User fill contact information, the <i>UC09</i> is performed	
		6. System executes UC09: Stores the user's contact information.
	7. Users confirm booking, the <i>UC10</i> is performed	
		8. System executes UC10: Validates and creates a pending booking
	9. User proceeds to payment, the <i>UC12</i> is performed	
		10. System executes UC12: Redirects to the payment process.
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
Post Condition	1. Users click any other pages hyperlink button.	
	2. User can return to PF step 1.	
	A pending booking is created, and the booking details are stored for confirmation.	

Table 4. 8. Use case specification of Choose Quantity (Source: Authors)

Use Case Name	Choose Quantity	Use Case Description

UC #	08	
Primary Actor	Registered user	
Use Case Story	User chooses the desired quantity of rooms to book for their stay.	User selects the number of rooms to book from the available options
Pre-Condition	User already login. Available rooms have been displayed based on the selected dates.	
Title: User choose desired rooms		
Primary Flow (PF)	Action Response	System Response
	1. User selected desire rooms	
		2. System updates the selected room and displays the corresponding price details If user change selection, go to AF1 If user confirm selection, go to next step
Alternative Flow (AF1)		3. System stores the chosen selected rooms and navigates to the next step in the booking process.
	Title: User Changes Selection	
	Action Response	System Response
Post Condition	1. User modifies the selected rooms.	
		2. System updates the selection and recalculates the price.
Post Condition	Selected rooms are recorded for the booking process	

Table 4. 9. Use case specification of Fill Contact Information (Source: Authors)

Use Case Name	Fill Contact Information	Use Case Description
UC #	09	
Primary Actor	Registered user	
Use Case Story	User fills in their contact information	User provides contact details necessary for completing the booking.
Pre-Condition	User already login. User has chosen the desired room.	
Title: User Fill Contact Information		
Primary Flow (PF)	Action Response	System Response
	1. User enters their full name, phone number, gender, date of birth, address, check-in-time and information of partner	
		2. System validates the input fields. If valid, go to next step If invalid, go to AF1
Alternative Flow (AF1)		3. System navigates to the next step in the booking process.
	Title: Valid Input Field	
	Action Response	System Response
	1. System detect missing required field include: - Full name - Gender	

	-Date of birth -Phone Number	
		2. System displays an error message
		3. System return user to PF step 1.
Post Condition	User's contact information is stored in the system for booking purposes.	

Table 4. 10. Use case specification of Confirm Booking (Source: Authors)

Use Case Name	Confirm Booking	Use Case Description	
UC #	10		
Primary Actor	Registed user		
Use Case Story	User confirms their booking details to finalize the pending booking	User reviews and confirms the booking details before proceeding to payment.	
Pre-Condition	User already login. User has entered all required contact information.		
Primary Flow (PF)	Title: User review and confirm accurate information		
	Action Response		
	1. User reviews personal information and room information.		
	2. Users confirm the booking If user want to make changes, go to AF1		

	If user does not want to make changes, go to next step	
		3. System creates a pending booking
Alternative Flow (AF1)	Title: User Changes Selected Room	
	Action Response	System Response
	1. User chooses to change the room information	
		2. System navigates user back to the room selection page
		3. System return user to PF step 1.
Post Condition	A pending booking is created.	

Table 4. 11. Use case specification of booking cancel (Source: Authors)

Use Case Name	Booking cancel	Use Case Description
UC #	11	
Primary Actor	Registed user	Registered user cancel their booking.
Use Case Story	Registered user cancel their booking	
Pre-Condition	User has a pending or done booking	
Primary Flow (PF)	Title: User cancel booking	
	Action Response	System Response
	1. Users select “Cancel” button	

		<p>2. System navigates User to “Booking History Page” showing booking details.</p> <p>If user select another page, go to AF1</p>
		<p>3. System valid booking status</p> <p>If valid (status is pending or done), go to next step</p> <p>If invalid, go to EF1</p>
	4. User confirms the cancellation.	
		<p>5. System updates the booking status to cancel and notifies the user of successful cancellation.</p>
Alternative Flow (AF1)		<p>Title: Click another page</p>
Exception Flow (EF1)	Action Response	System Response
	1. Users click any other pages hyperlink button.	
Exception Flow (EF1)	2. User can return to PF step 1.	
	Title: Verified Booking Status	
	Action Response	System Response
Post Condition	1. System detects that Booking Status is Cancel	
		2. System displays an error message.
		3. System return user to PF step 1.
Post Condition	The booking status is updated to cancel , and the user is notified of the successful cancellation.	

Table 4. 12. Use case specification of payment (Source: Authors)

UC #	12	
Primary Actor	Registered user	This use case allows a registered user to process payment for a confirmed booking and receive a booking confirmation along with a receipt.
Secondary Actors	Payment gateway system	
Use Case Story	Registered user payment for booking rooms	
Pre-Condition	User has confirmed the booking information	
Primary Flow (PF)	Title: User paying selected room	
	Action Response	System Response
	1. The user selects the "Continue" button in page confirm information.	
		2. System calculates the total amount based on the selected room details and navigates the user to the "Payment page" If user select another page, go to AF1 If user select cancel button, go to AF2
	3. User scan QR and confirms the payment.	
		4. System processes the payment through the payment gateway If user takes too long time to make payment, go to AF3

		5. System updates the booking status to “done” and sends a booking confirmation
Alternative Flow (AF1)	Title: Click another page	
	Action Response	System Response
	1. Users click any other page's hyperlink button.	
	2. User can return to PF step 1.	
Alternative Flow (AF2)	Title: Click cancel payment	
	Action Response	System Response
	1. User click cancel button.	
		2. System displays a confirmation message asking if the user wants to cancel the payment.
	3. User confirms the cancellation.	
		4. System sends notification successfully cancel payment and navigates the user to the “Confirm Information Page”
	5. User does not confirm the cancellation.	
Alternative Flow (AF3)		6. System return user to PF step 3.
	Title: Session Timeout	
Alternative Flow (AF3)	Action Response	System Response
	1. User takes too long to complete the payment (limit 15 minutes)	

		2. System displays an notification message.
		3. System return user to PF step 1.
Post Condition	The payment is processed successfully, the booking status is updated to ‘Done’, and the booking confirmation and receipt are sent to the user.	

Table 4. 13. Use case specification of Logout (Source: Authors)

Use Case Name	Logout		Use Case Description
UC #	14		
Primary Actor	Registered user		
Secondary Actor	Admin		This use case allows a registered user or admin to logout from the system.
Use Case Story	Logout account from website		
Pre-Condition	User current login Admin current login		
Primary Flow (PF)	Title: Logout		
	Action Response		System Response
	1. User /Admin select “Logout” button		
			2. The system displays confirmation form.
	3. User/Admin select confirmation to logout		

	If confirm to logout, go to next step	
		4. System their cookies and navigate to main page (User) / login page (Admin)
Post Condition	User logout successfully	

Table 4. 14. Use case Manage Users (Source: Authors)

Use Case Name	Manage Users	Use Case Description
UC #	15	
Primary Actor	Admin	
Use Case Story	Admin can create, read, update, and delete users information.	This use case enables the Hotel Administrator to manage users by creating, reading, updating, and deleting users information.
Pre-Condition	The user successfully logs in as the Administrator. The system has verified the user's role and permissions.	
Primary Flow (PF)	Title: Admin manage user information	
	Action Response	System Response
	1. Admin selected activity “Create User”	
		2. System executes AF1: Create a user alternative flow is performed
	3. Admin selected activity “Read User Information”	
		4. System executes AF2: Read User alternative flow is performed.

	5. Admin selected activity “Update User Information”	
		6. System executes AF3: Update User alternative flow is performed.
	7. Admin selected activity “Delete User”	
		8. System executes AF4: Delete Room alternative flow is performed.
Title: Create User		
Alternative Flow (AF1)	Action Response	System Response
		1. System displays the user creation screen.
	2. Admin enters and saves the user details, including name, username, email, password.	
Alternative Flow (AF2)		3. The system validates the input and saves the user information in the database [EF1].
	Title: Read User	
	Action Response	System Response
Alternative Flow (AF3)		1. System displays a list of users.
	2. Admin selects a user to view its detailed information.	
Title: Update User		
Alternative Flow (AF3)	Action Response	System Response
		1. System displays the user update screen

	2. Admin enters and saves the user details	
		3. The system validates the input and saves the user information in the database [EF1].
Alternative Flow (AF4)	Title: Delete User	
	1. Admin selects a specific user to delete.	
		2. System removes the user information from the database
Exception Flow (EF1)	Title: Invalid user information	
		1. System notifies the error to admin.
	2. Admin re-enter the details, restart, or terminate the use case	

Table 4. 15. Use case Manage Room (Source: Authors)

Use Case Name	Manage Room	Use Case Description
UC #	16	
Primary Actor	Admin	
Use Case Story	Admin can create, read, update, and delete room information.	This use case enables the Hotel Administrator to manage rooms by creating, reading, updating, and deleting room information.
Pre-Condition	The user successfully logs in as the Administrator. The system has verified the user's role and permissions.	
Primary Flow (PF)	Title: Admin manage room information	
	Action Response	System Response
	1. Admin selected activity “Create a Room”	

		2. System executes <i>AF1</i> : Create Room alternative flow is performed
	3. Admin selected activity “Read Room Information”	
		4. System executes <i>AF2</i> : Read Room alternative flow is performed.
	5. Admin selected activity “Update Room Information”	
		6. System executes <i>AF3</i> : Update Room alternative flow is performed.
	7. Admin selected activity “Delete a Room”	
		8. System executes <i>AF4</i> : Delete Room alternative flow is performed.
Alternative Flow (AF1)	Title: Create a room	
	Action Response	System Response
		1. System displays the room creation screen.
	2. Admin enters and saves the room details, including room type, capacity, price, and other specifications.	
		3. The system validates the input and saves the room information in the database [EF1].
Alternative Flow (AF2)	Title: Read room	
	Action Response	System Response
		1. System displays a list of rooms.
	2. Admin selects a specific room to view its detailed information [EF2].	
	Title: Update room	

	Action Response	System Response
Alternative Flow (AF3)		1. System displays the room update screen
	2. Admin enters and saves the room details, including room type, price, and status.	
		3. The system validates the input and saves the room information in the database [EF1].
Alternative Flow (AF4)	Title: Delete room	
	1. Admin selects a specific room to delete.	
		2. System removes the room information from the database
Exception Flow (EF1)	Title: Invalid room information	
		1. System notifies the error to admin.
Exception Flow (EF2)	2. Admin re-enter the details, restart, or terminate the use case	
	Title: Room information cannot be accessed.	
		1. System notifies the error to admin.
	2. Admin, restart, or terminate the use case	

Table 4. 16. Use case Manage Bookings (Source: Authors)

Use Case Name	Manage Bookings	Use Case Description
UC #	17	This use case enables the Hotel Administrator to manage bookings by creating, reading, updating, and deleting booking information.
Primary Actor	Admin	
Use Case Story	Admin can create, read, update, and delete bookings information.	

Pre-Condition	The user successfully logs in as the Administrator. The system has verified the user's role and permissions.	
Primary Flow (PF)	Title: Admin manage booking information	
	Action Response	System Response
	1. Admin selected activity “Read booking Information”	
		2. System executes <i>AF1</i> : Read Booking alternative flow is performed.
	3. Admin selected activity “Update Booking Information”	
Alternative Flow (AF1)		4. System executes <i>AF2</i> : Update Booking alternative flow is performed.
	Title: Read Booking	
	Action Response	System Response
Alternative Flow (AF2)		1. System displays a list of Bookings.
	2. Admin selects a specific booking to view its detailed information [EF2].	
Alternative Flow (AF2)	Title: Update Booking	
	Action Response	System Response
	2. Admin enters and saves the booking details, including check-	1. System displays the booking update screen

	in-date, check-out-date, price, status	
		3. The system validates the input and saves the booking information in the database [EF1].
Exception Flow (EF1)	Title: Invalid booking information	
	2. Admin re-enter the details, restart, or terminate the use case	1. System notifies the error to admin.
Exception Flow (EF2)	Title: Booking information cannot be accessed.	
	2. Admin, restart, or terminate the use case	1. System notifies the error to admin.

4.4. Class Diagram

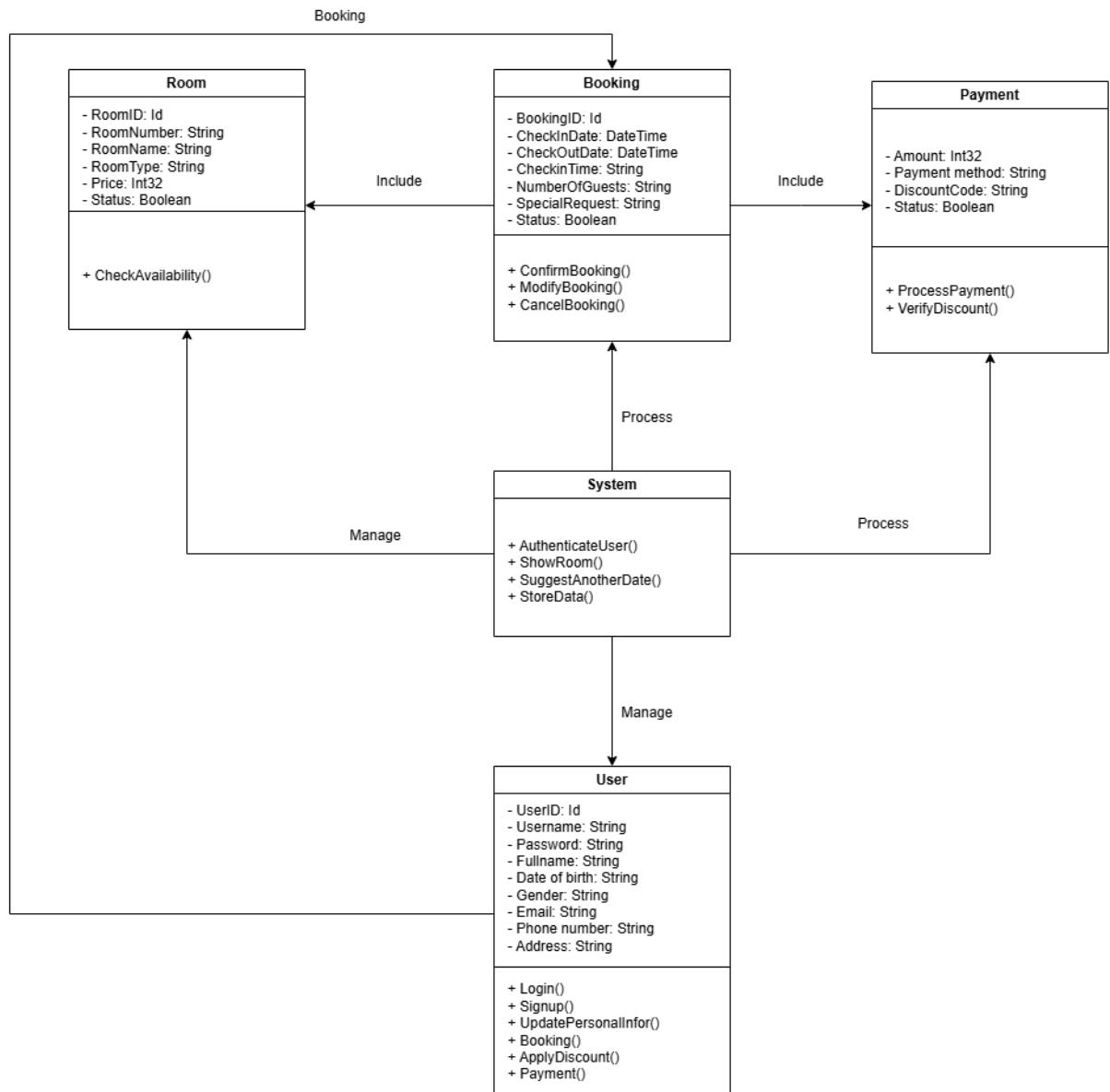


Figure 4. 2. Class Diagram (Source: Authors)

The Class Diagram above describes the main classes and their relationships in the hotel booking process:

- **User**: Represents the user with personal attributes and account-related methods such as creating an account, logging in, editing personal information and booking methods, selecting discount codes and payment.
- **Room**: Describes information about the rooms in the hotel.

- Booking: Handles the booking process, including confirming the booking, editing booking information or canceling the booking.
- Payment: Manage the payment process, including applying discount codes and processing payments.
- System: Represents the overall system, managing all the processes and interactions between other layers.

4.5. Sequence Diagram

Sequence Diagram for Register

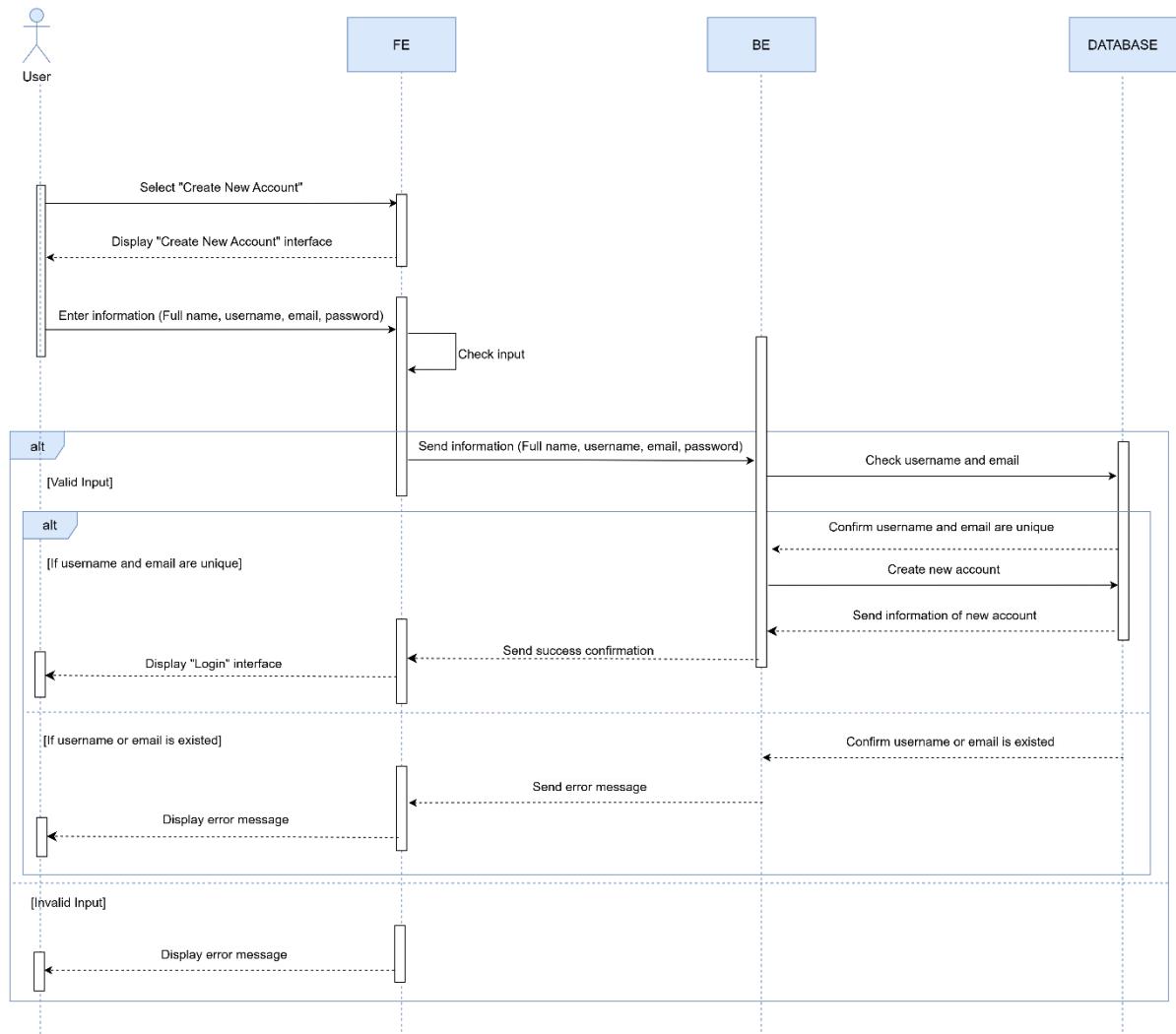


Figure 4. 3. Sequence Diagram for Register (Source: Authors)

For the Registration process, the user will select "Create new account" and the Frontend will display the "Create new account" interface. Then, the user enters the

necessary information such as Full Name, Username, Email and Password to create an account. The Frontend will check the entered information. If the information is valid (no syntax errors, characters, length,...), it will send this information to the Backend and will check if the Username and Email that the user entered already exist or not. If Username or Email does not exist yet, a new account will be created in the Database and the User Interface will redirect to the login page so the user can log in to the system. Conversely, if the Username or Email already exists, the user will receive a notification.

Sequence Diagram for Login

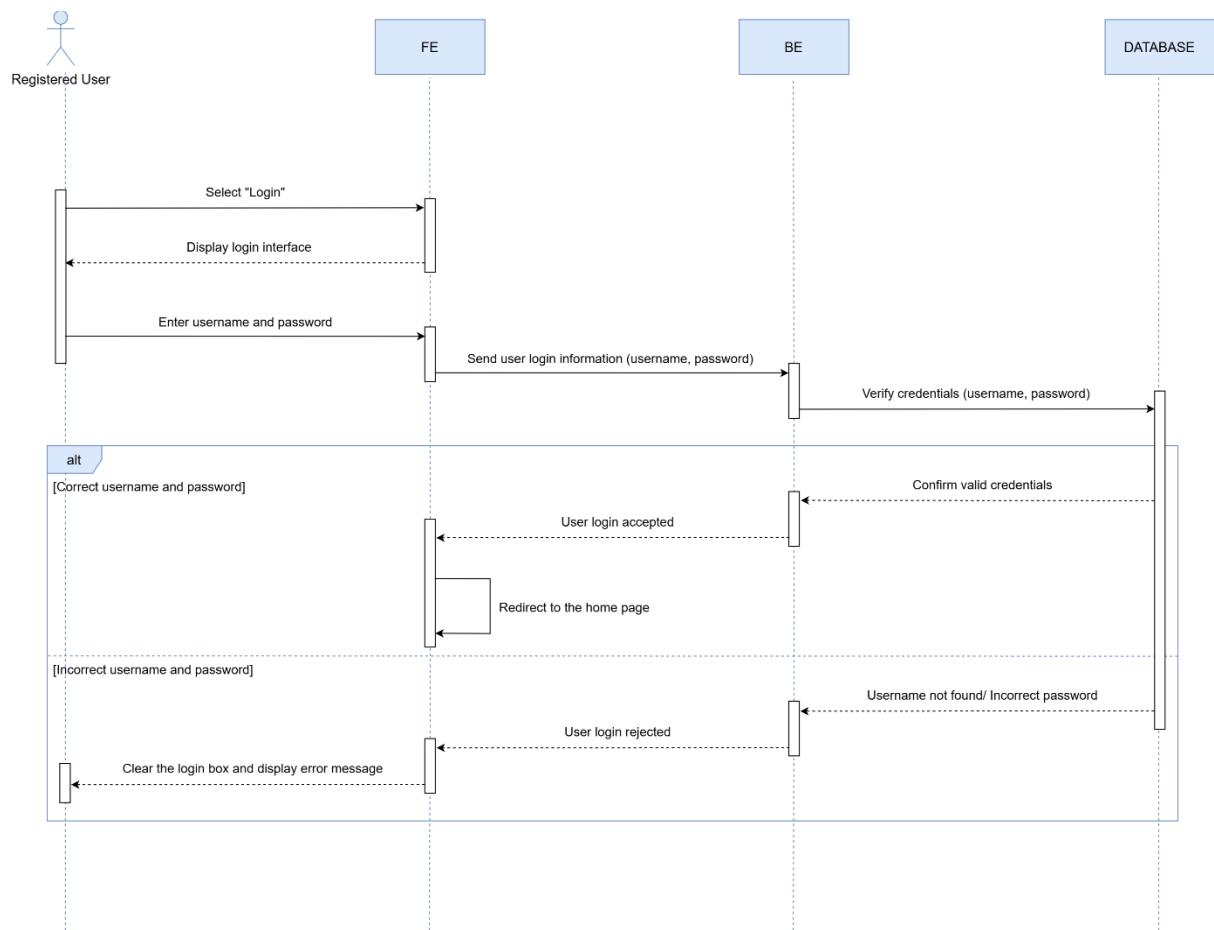


Figure 4. 4. Sequence Diagram for Login (Source: Authors)

The Login process is intended for users who have already registered an account. Frontend will display the Login interface when the user requests it. Then, the user will enter account information including Username and Password. Frontend will pass this information to Backend for confirmation. If the information is valid (both Username

and Password are correct), the user will be allowed to log in and the Frontend will return to the home page. If the information is invalid (Username or Password is incorrect), the Backend will not allow the user to log in and the Frontend will delete the information and display a message to the user.

Sequence Diagram for Booking

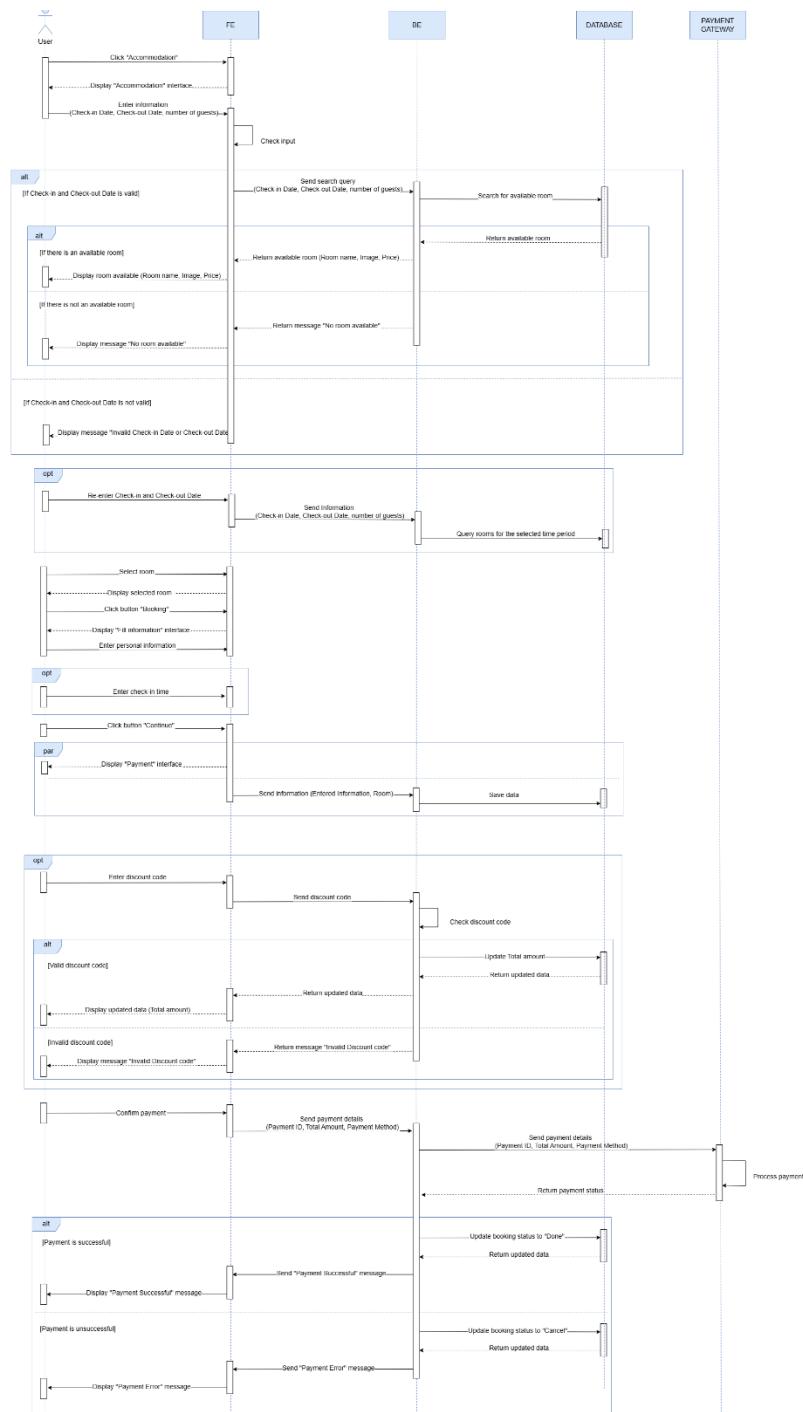


Figure 4. 5. Sequence Diagram for Booking (Source: Authors)

The booking process begins when the user selects “Accommodation.” The Frontend will display an interface where the user can fill in the required information, including “Check-in Date, Check-out Date, and the number of guests.” The Frontend will validate the entered information. If the information is invalid, a notification will be displayed to the user. If the information is valid, it will be sent to the Backend, which will use this data to search for available rooms in the Database.

If there are available rooms, the Database will send the room information to the Backend, which will forward details (Room name, Image, Price) to the Frontend. The Frontend will then display the available rooms on the interface. If no rooms are available, a notification saying “No room available” will be shown to the user, who can modify the Check-in and Check-out dates to search again.

Once the Frontend displays the available rooms, the user selects the desired room and clicks the “Booking” button. At this point, the Frontend will display an interface for the user to input their personal details. The user can also specify a preferred check-in time if needed.

To proceed, the user clicks the “Continue” button. The Frontend will send the entered personal details along with the selected room information to the Backend, which will save this data in the Database. Simultaneously, a Payment interface will be displayed for the user to complete the payment process.

The user may enter a discount code if applicable, and this code will be sent to the Backend for validation. If the code is invalid, a message saying “Invalid Discount Code” will be displayed. If the code is valid, the “Total Amount” will be updated. The updated Total Amount will be displayed on the interface.

After the user confirms payment, the payment details will be sent to the Payment Gateway, which will process the payment and return the result to the Backend. The Backend will then update the booking status in the Database accordingly: “Done” for successful payment or “Cancel” for unsuccessful payment.

If the payment is successful, a notification saying “Payment Successful” will be displayed to the user. If the payment fails, a message saying “Payment Error” will be shown.

4.6. Activity Diagram

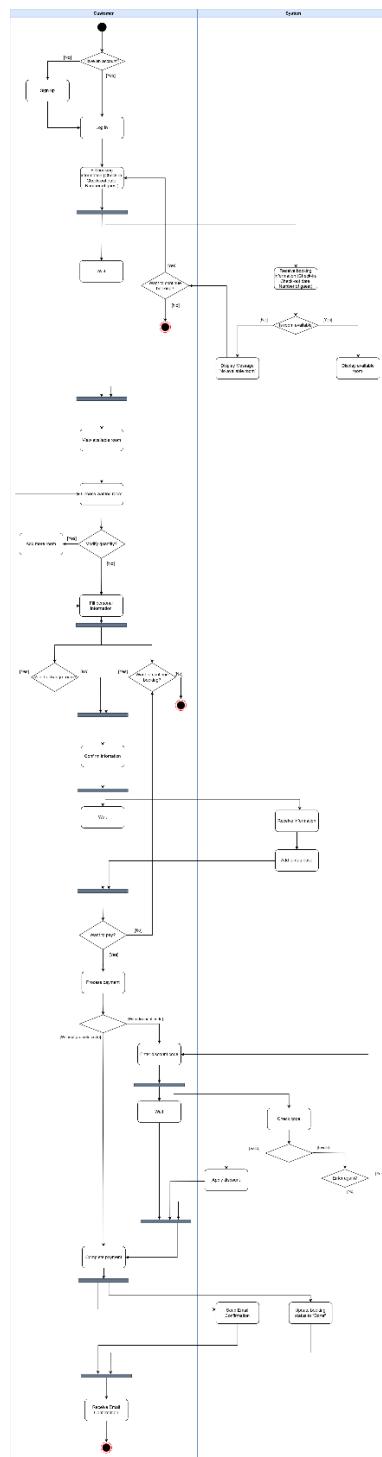


Figure 4. 6. Activity Diagram for Booking (Source: Authors)

This activity diagram illustrates the "Booking" function, describing the process flow and activities when a User makes a booking. The diagram provides a structured view of the steps from entering information to confirming payment. In this process, there are two primary entities: the "User" and the "System," each performing different tasks.

To make a booking, the User must log into the System. If the User does not have an account, they can register one. After logging in, the User fills in information including Check-in Date, Check-out Date, and Number of Guests to search for suitable rooms. The System then checks the availability of rooms within the dates specified by the User. If no rooms are available, a “No available room” message is displayed, and the User can re-enter the information if they wish to continue. If there are available rooms, the System will display them to the User.

The User selects the desired room and can adjust the number of rooms if needed. If no adjustments are necessary, they proceed to enter their personal information. After completing personal details, if the User wishes to continue with the booking without changing rooms, they confirm the information. The information confirmed by the User is then sent to the System and added to the database.

The User can complete payment with or without a discount code. If using a discount code, the System checks its validity. If valid, the code is applied. Afterward, the User completes the payment. At this point, the System sends a confirmation email to the User and updates the booking status to “Done”. Once the User receives the confirmation email, the process ends.

CHAPTER 5: USER INTERFACE DESIGN AND DATA DESIGN

5.1. User interface design

5.1.1. Overview and purposes

This chapter focuses on User Interface Design, a vital aspect of creating engaging and effective digital products. The primary purpose of this chapter is to explore the principles and methodologies that guide the design of user interfaces, ensuring they are intuitive, visually appealing, and aligned with user needs. Specifically, we use Figma to visualize the new interface, aiming to give readers a comprehensive understanding of our proposed functions from a UX-UI perspective.

- Enhancing User Interaction: Aim to create interfaces that facilitate seamless interactions between users and the system, improving overall user satisfaction and engagement.
- Visual Communication: Emphasize the importance of visual elements in conveying information clearly and effectively, enabling users to navigate and utilize the application efficiently.
- Prototyping for Feedback: Highlight the significance of developing prototypes to gather user feedback early in the design process, allowing for iterative improvements that align with user expectations.

5.1.2. Deliverables

The deliverables for this chapter encompass several key tasks essential for effective user interface design. These tasks include:

- Sitemap: Develop a sitemap that outlines the structure of the application, illustrating how different screens and sections are organized and interconnected. This visual representation aids in understanding user navigation paths.

- Screen Mockups and Prototype: Create detailed screen mockups that visualize the layout and design elements of the interface. Additionally, develop interactive prototypes that simulate user interactions, allowing for testing of design concepts and gathering valuable user feedback.

5.1.3. Sitemap

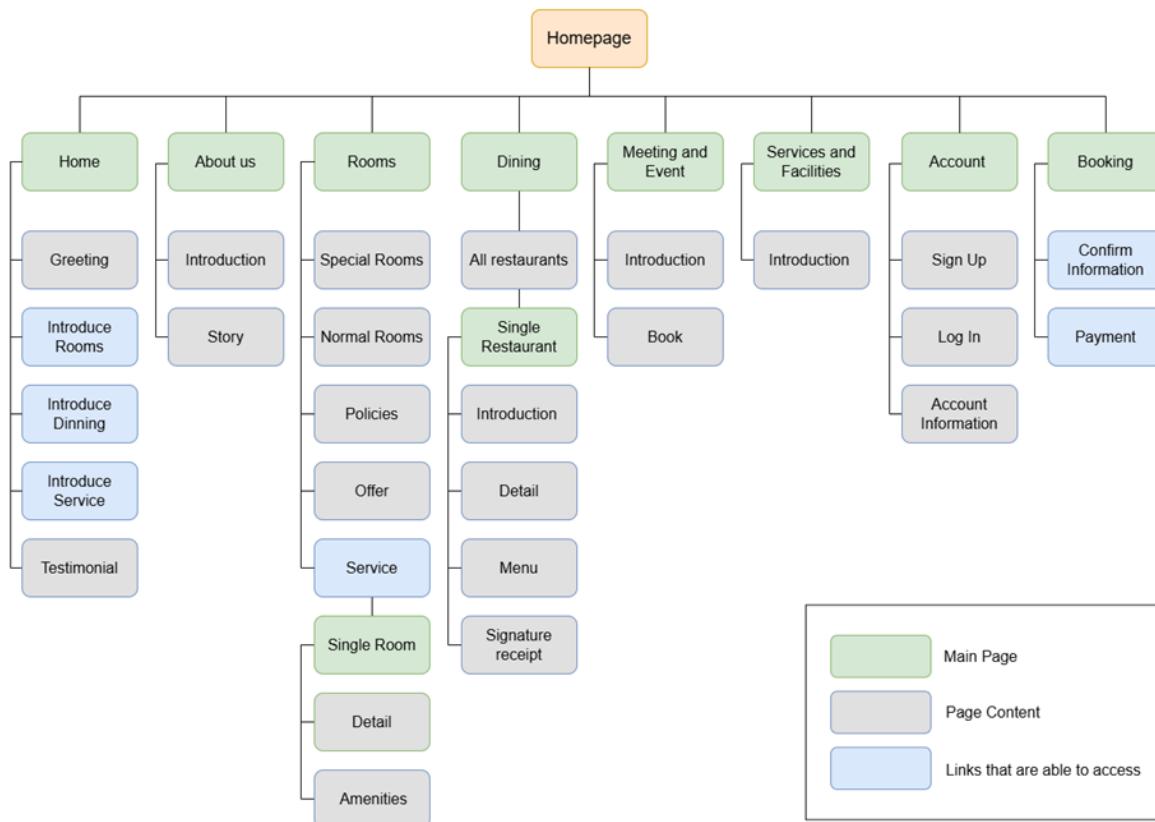


Figure 5. 1. Sitemap (Source: Authors)

Home: This page provides an overview of the main components of the website, including greetings, introductions to rooms, dining options, services, and customer testimonials.

About us: Provides information about the hotel's history and brand story. It also gives an overall view of the hotel's historical values.

Rooms: Offers information about the types of rooms available in the hotel, including detailed descriptions for each room type and related services.

Dining: Features various restaurant styles, offering cuisines ranging from European to Asian.

Meeting and Event: Helps customers find information about meeting and event booking services, with rooms of various capacities.

Services and facilities: Introduces amenities such as the gym and other available services.

Account: Manages user accounts with functions such as sign-up, log-in, and updating personal information.

Booking: Include Booking Detail Confirmation and Payment

5.1.4. Screen Mockups and Prototype

5.1.4.1 User

Header



Figure 5. 2. Header interface of the website (Source: Authors)

Main Sections of the “Dé Luna Hotel” Website Menu:

- Home: Provides highlighted information, giving a first impression of the hotel and an overview of services.
- Sign up/Log in: Users will click on one of these buttons for create account and log in for the booking process
- Overview: Detailed introduction to the hotel, sharing the brand story and unique values.
- Rooms: Displays room types, illustrative images, details on size and layout, helping guests choose rooms easily.

- Dining: Showcases culinary services, special menus, and unique dining experiences offered by the hotel.
- Meetings & Events: Information on conference spaces, event venues, and professional event packages.
- Services & Amenities: Introduces additional amenities like a gym, swimming pool, and services that enhance the guest experience.
- Right Corner of the Navigation Bar: Includes Log In, Create Account buttons, and icons for user account, designed simply and elegantly, matching the primary beige tone for easy user access.

Footer



Figure 5. 3. Footer interface of the website (Source: Authors)

The footer is structured with three columns: Reservation, Hotel, and About Us.

- Reservation: Provides essential contact details for booking and inquiries.
 - + Front Desk: Customers can directly contact the hotel's front desk in Ho Chi Minh City via hotline, address, and email for quick booking and inquiries.
 - + Hanoi Office: Contact details for the Hanoi representative office, including phone, address, and email, making it convenient for customers in the northern region.
- Hotel: Contains quick links to main sections of the website: "Home," "Overview," "Rooms," "Dining," "Meetings & Events," and "Services & Amenities," giving a comprehensive view of the hotel's offerings.

- About Us: Features a link to "Overview," introducing the hotel's background, values, and mission to customers.

Login

The login page has a light blue header with the text "Đăng nhập". Below it is a white form area. It contains two input fields: "Tên tài khoản" (Account name) and "Mật khẩu" (Password). To the right of the password field is a link "Quên mật khẩu ?" (Forgot password?). Below the inputs is a large blue "Đăng nhập" (Login) button. At the bottom of the form, there is a small note: "Nếu bạn chưa có tài khoản, hãy ấn Đăng ký" (If you don't have an account, click Register).

Figure 5. 4. The login page interface of the website (Source: Authors)

If users forget their registered account password, they can click on “Forgot Password” to reset it. Please enter the registered email address to receive a new password.

The password reset page has a light blue header with the text "Đặt lại mật khẩu" (Reset password). Below it is a white form area. It contains two input fields: "Địa chỉ email" (Email address) and "Nhập lại mật khẩu mới" (New password). Below the inputs are two buttons: a blue "Gửi" (Send) button on the left and a blue "Đăng nhập" (Login) button on the right. A note above the "Gửi" button says: "Vui lòng nhập địa chỉ email bạn đã đăng ký để nhận được email đặt lại mật khẩu" (Please enter the email address you registered to receive a password reset email).

Figure 5. 5. The password reset interface of the website (Source: Authors)

After clicking the submit button, the system will notify the user that the password has been successfully reset, and they can proceed to log in.

Signup

If users do not have an account, they can register through the registration form. Users need to fill in all required information, such as full name, username, email address, password, and then click the "Register" button.

Figure 5. 6. The signup page interface of the website (Source: Authors)

My Account

Figure 5. 7. Account & Edit Information Tab (Source: Authors)

This is the user's personal information page. Here, users can edit their details such as full name, gender, date of birth and phone number.

Home

The Home page includes an overview of the hotel as well as some news updated to the users. In this page, the management board tries to convey to customers the quality services, amenities and wonderful experiences that the hotel brings to customers, including style and spacious atmosphere, ideal for parties and special events.

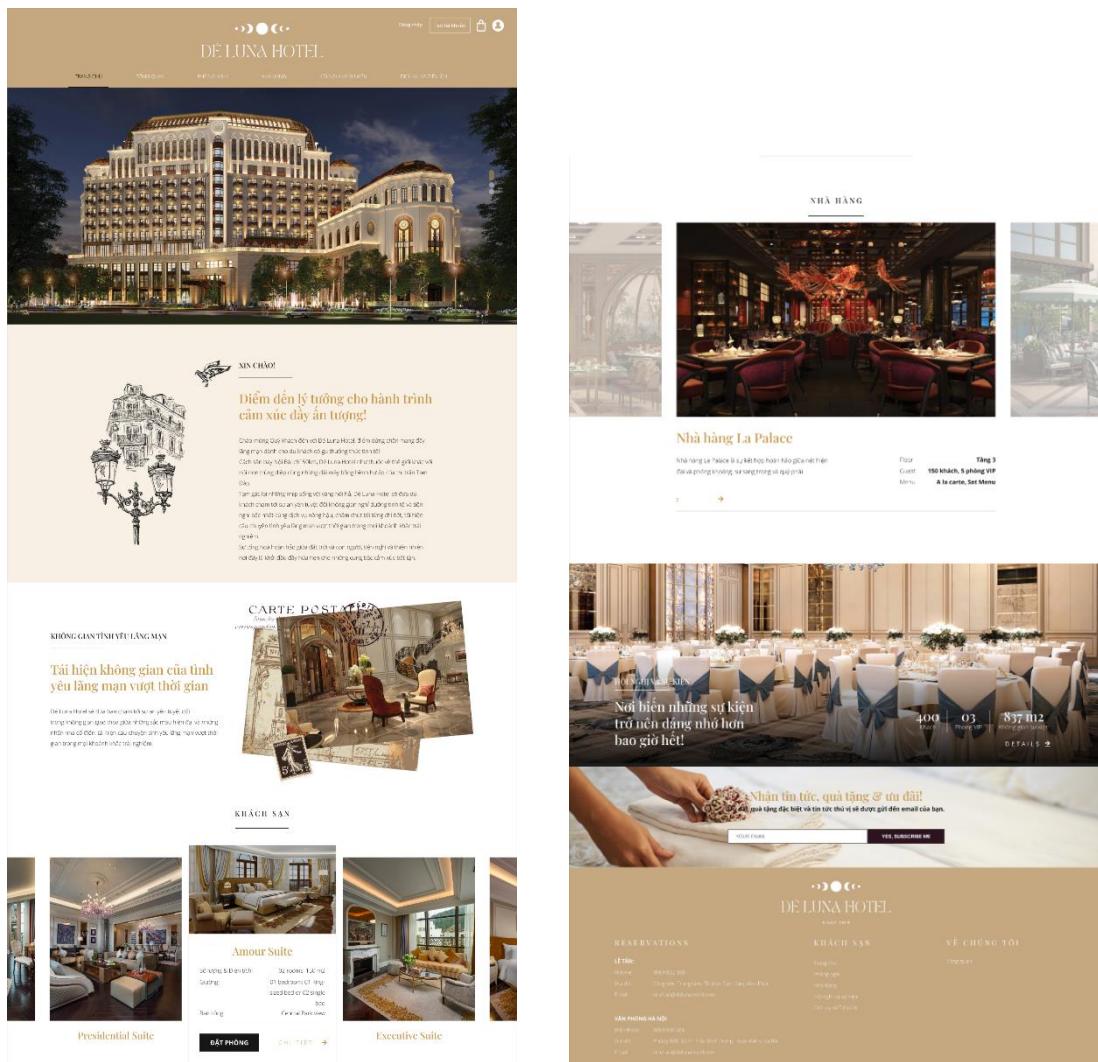


Figure 5. 8. Home interface of the website (Source: Authors)

About us

The page introduces the hotel's romantic and luxurious atmosphere, inspired by timeless charm. Sections emphasize the hotel's sophisticated design, upscale ambiance, and attention to detail. Notably, the page describes a peaceful location with scenic views, creating a relaxing environment for guests. It also provides information on distances to nearby attractions and concludes with a subscription section for news and special offers.

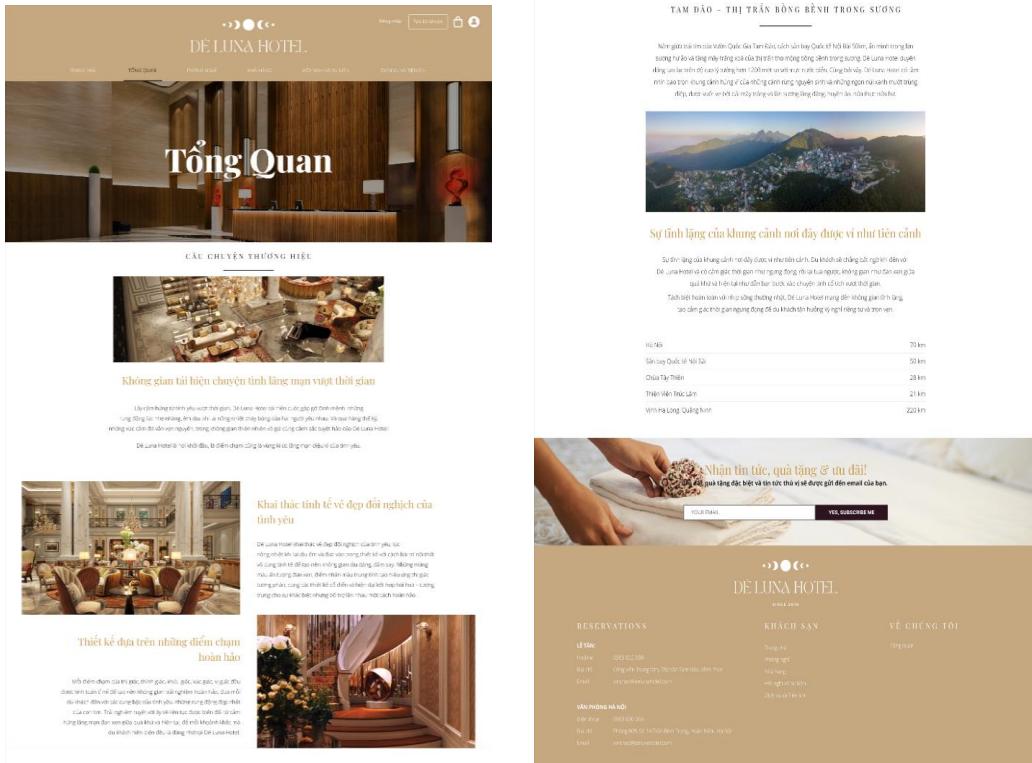


Figure 5. 9. About us interface of the website (Source: Authors)

Room

The hotel's room page displays all available room types, complete with illustrative images, detailed descriptions, and booking options for each. At the top, customers can use the search bar to select dates and the number of guests, making the search process quick and convenient. The room list is organized clearly with information on capacity, amenities, and a "Book" button for each option.

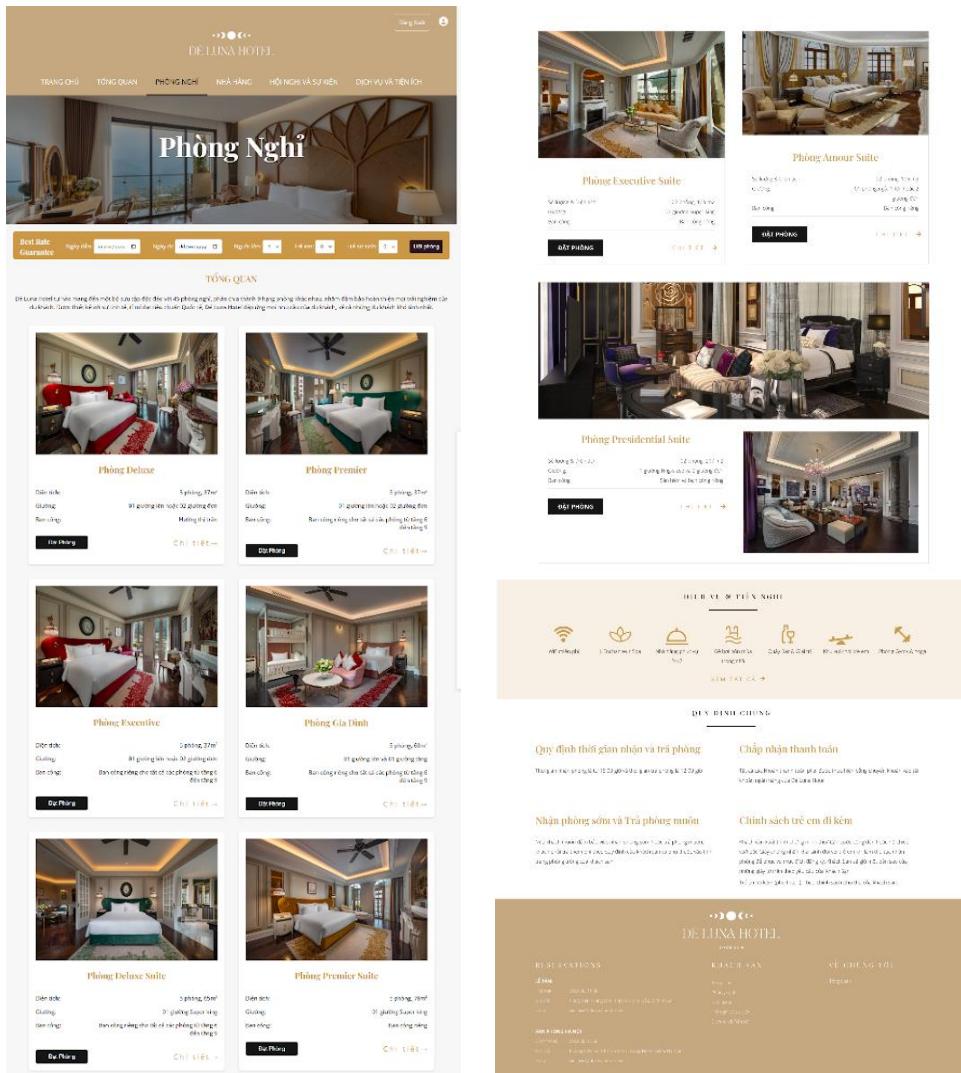


Figure 5. 10. Room interface of the website (Source: Authors)

After clicking "Book Now," a booking interface appears, allowing customers to select dates, specify the number of guests, and enter personal information. In addition, to view detailed information about the rooms, customers can click the View details button, the system will display the detailed information page as well as the amenities and services of that room.

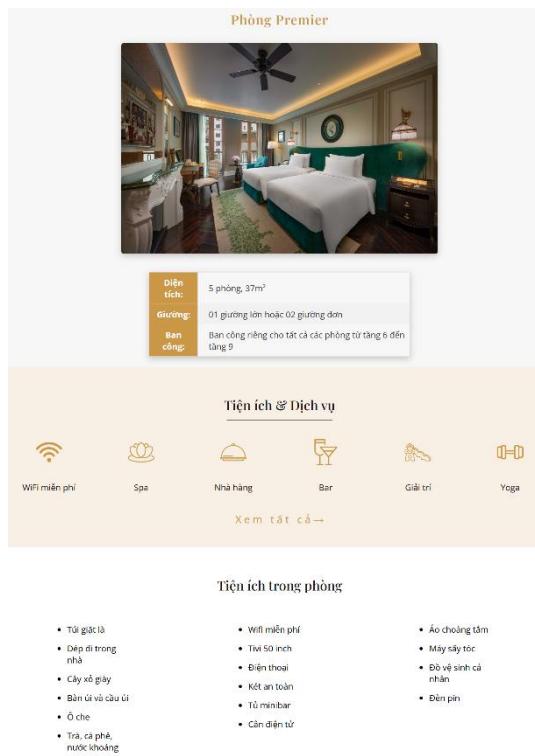


Figure 5. 11. Room details (Source: Authors)

Booking

The booking interface shows search parameters at the top: 'Best Rate Guarantee', dates from '06/11/2020' to '08/11/2020', guests for '2', children for '0', and babies for '0'. Below this, a 'Thay đổi' (Change) button and sorting options ('Lọc theo giá', 'Chọn khoảng giá', 'Sắp xếp theo giá', 'Chọn cách sắp xếp') are visible. The main area is titled 'Danh sách phòng trống' (List of available rooms) and lists six room types with their prices: Deluxe Hollywood (2240000), Deluxe Twins (3140000), Deluxe King (3140000), Deluxe King (3140000), and Executive King (3740000). To the right, a summary box titled 'Hóa đơn' (Bill) shows a total of 6480000 VND and a large 'Đặt phòng' (Book) button.

Figure 5. 12. Booking list (Source: Authors)

Based on the date information that the customer has entered the booking bar, the system will display a list of available rooms during this period. Here, customers can view room information including room photo, room name and price.

When customers choose rooms, the prices of the rooms will be updated in the invoice on the right. In addition, to facilitate room selection, customers can choose to Filter by price or Sort by price.

The screenshot shows a booking confirmation interface. At the top, there are tabs for 'Xác nhận thông tin' (Confirm information) and 'Thanh toán' (Payment). A dropdown menu for currency conversion is visible. The main area is divided into sections:

- Xác nhận thông tin**: Contains fields for contact information (Name, Gender, Date of Birth, Phone, Email, Address), guest information (Number of guests, Check-in/Check-out dates, Guest names), and room selection (Room type, Room name, Price).
- Hóa đơn**: Shows an invoice summary with a total amount of 6,480,000 VND.

Figure 5. 13. Confirm information (Source: Authors)

After selecting a room, customers will confirm information including contact information, guest information, and room information. If the information is wrong, customers can edit it directly on the information bars. In addition, customers can choose room function to Delete room or Add another room. After confirming the general information, customers will review the invoice section, where they can enter the existing discount code. In addition, customers can choose to display prices in different currencies using the website's automatic exchange rate update function.

When sure about the information, the customer presses the Next button to go to the Payment screen.

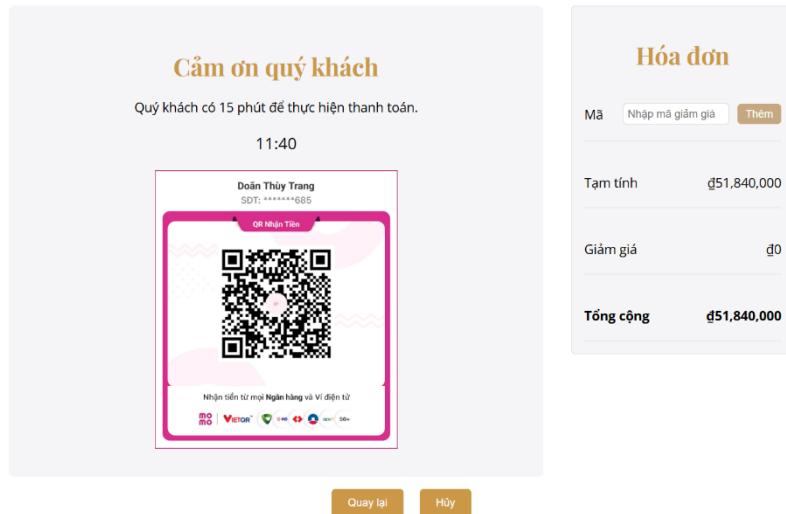


Figure 5. 14. Payment (Source: Authors)

The payment screen includes the payment code and invoice. The system displays a 15-minute countdown for customers to pay; After 15 minutes, if the customer has not made payment or the payment has failed, the system will display a notification and return to the Confirm information screen, asking the customer to redo the operations. In addition, customers can proactively cancel payment or return to the Confirm information step using the function buttons on the interface.

Dining

The "Dining" page of De Luna Hotel presents a collection of restaurants with diverse styles and culinary experiences, from the French elegance of La Vie en Rose, the cozy ambiance of Les Amoureux, to the panoramic views of Le Ciel. Each restaurant offers a unique flavor and atmosphere, catering to the refined tastes of guests.

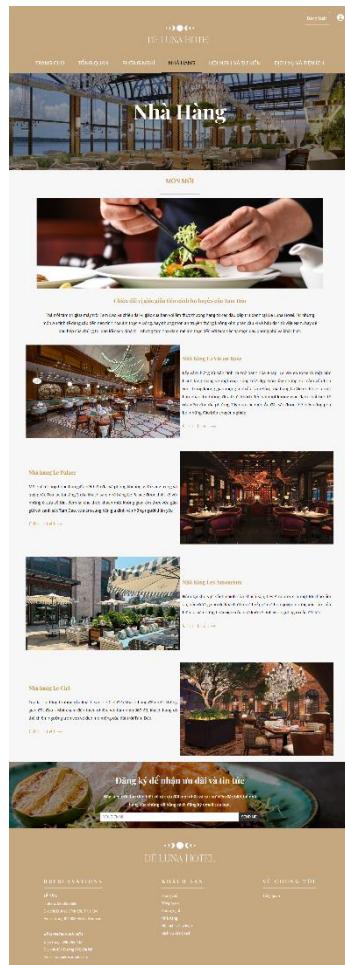


Figure 5. 15. Dinning interface of the website (Source: Authors)

When selecting "Details" for the restaurant, users will see comprehensive information about the restaurant, including its location, seating capacity, and service hours for breakfast, lunch, and dinner. The page also provides a diverse menu ranging from buffet options to à la carte dishes, accompanied by images of featured dishes and a contact option for reservations.

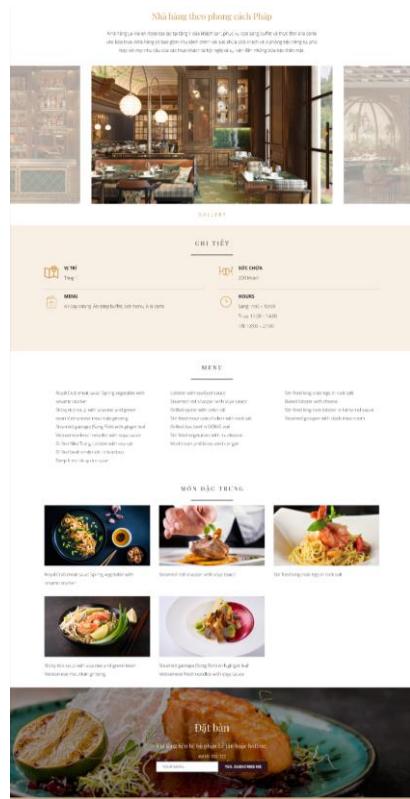


Figure 5. 16. Dining detail interface of the website (Source: Authors)

Meeting

The De Luna Hotel website showcases its Conference & Event services with an elegant design and detailed information on event amenities. The Overview section highlights professionalism and commitment to memorable events. The hotel offers various meeting rooms, with the standout being the Amour Banquet Room, which provides multiple seating arrangements like Classroom, Theatre, and U-Shape, accommodating from 120 to 440 guests. Additionally, there are three functional meeting rooms – Amour 1, Amour 2, and Amour 3 – designed flexibly to cater to different types of events. The Tea Break Area offers a comfortable resting space for attendees, while Eden Garden provides a unique outdoor event space for an exceptional experience. At the bottom of the page, there is a consultation form and contact information to help customers easily connect and receive assistance.

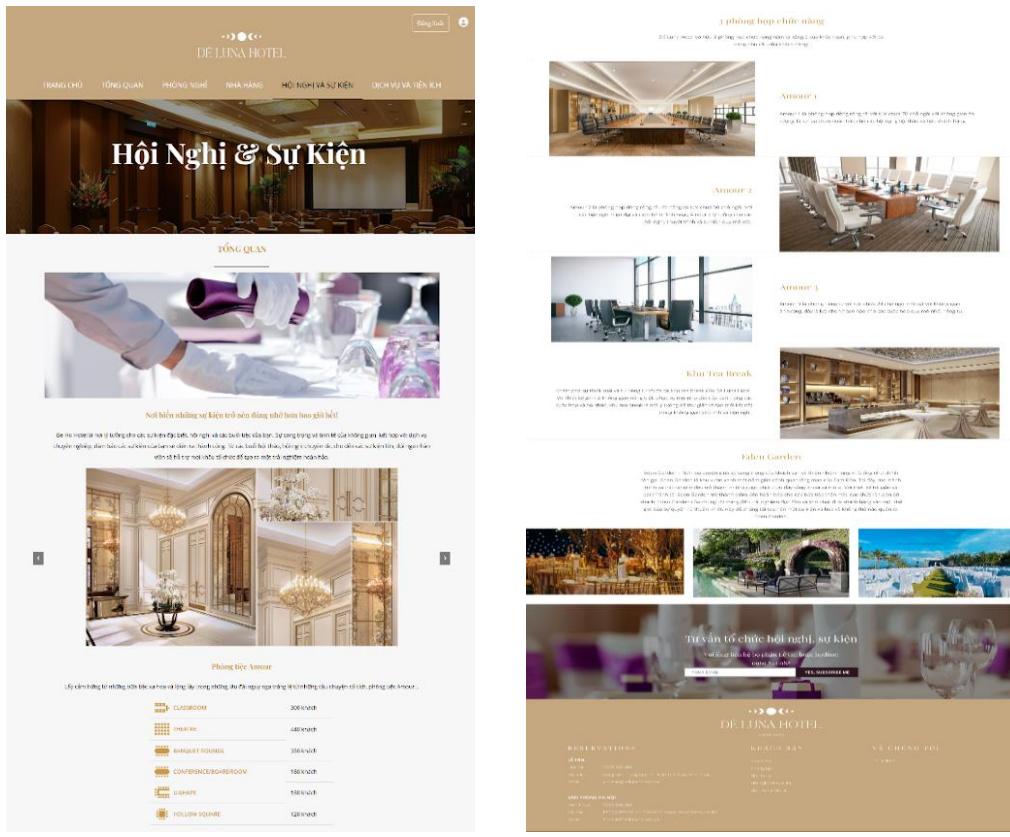


Figure 5. 17. Meeting interface of the website (Source: Authors)

Service

The "Services & Amenities" page of De Luna Hotel provides guests with a comprehensive view of the premium facilities, from the all-season swimming pool to the modernly equipped Gym & Yoga club. The elegant design and detailed content make it easy for guests to explore and plan their relaxing getaway.

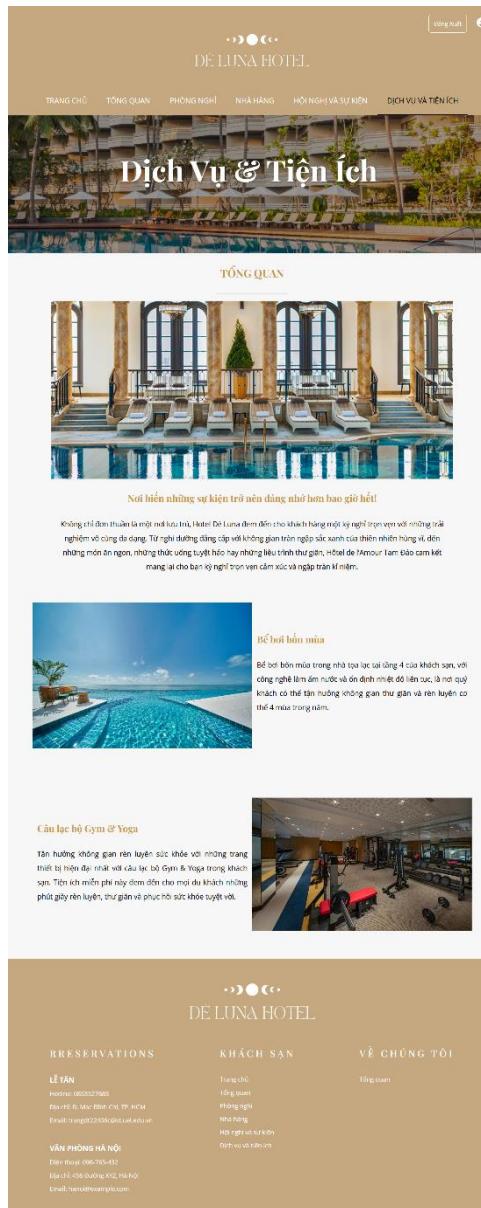


Figure 5. 18. Service interface of the website (Source: Authors)

5.1.4.2 Admin

Admin Login Page

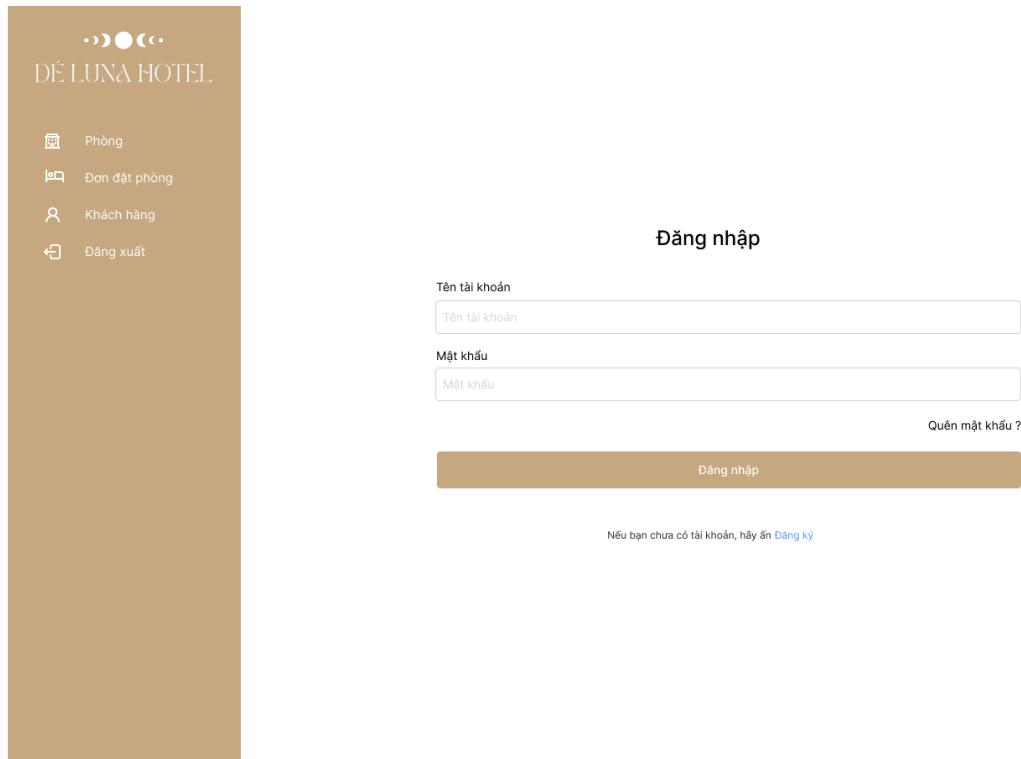
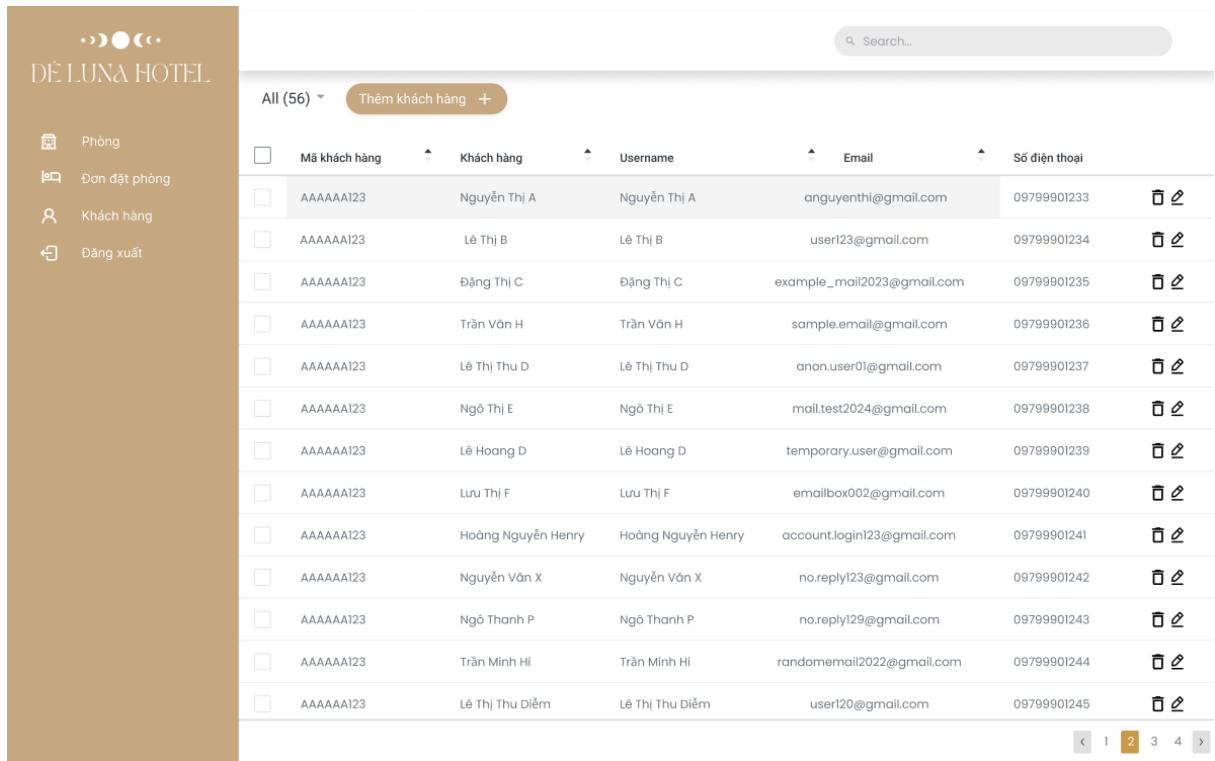


Figure 5. 19. The admin login page interface (Source: Authors)

The login page provides the required fields to access the site, including fields for username and password, along with the page footer.

Customer Management Page

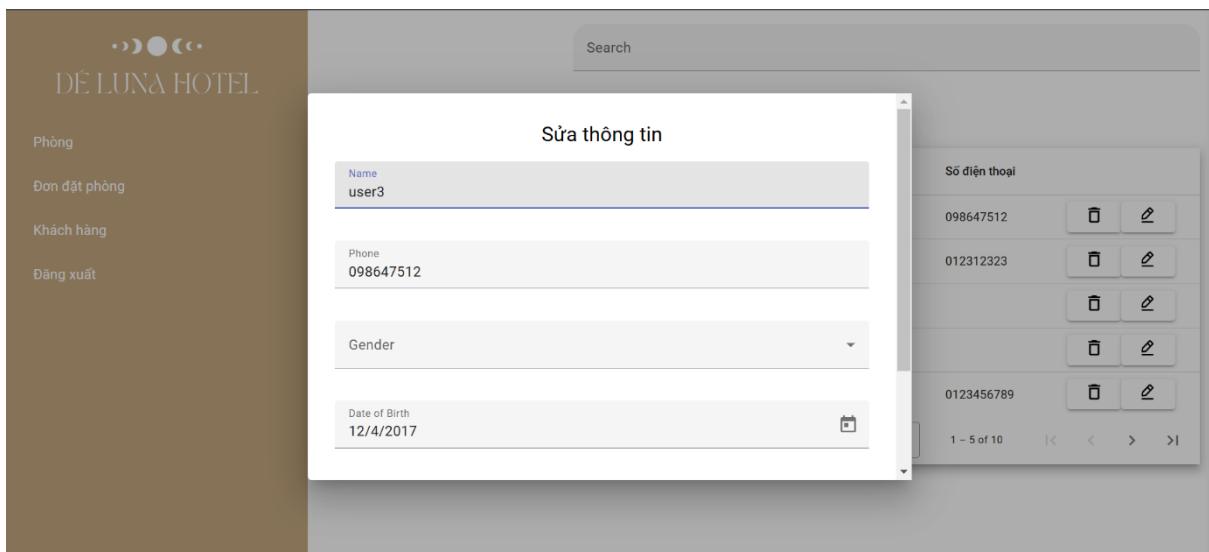


The screenshot shows a web-based admin interface for managing customer data. On the left, there's a sidebar with navigation links: Phòng, Đơn đặt phòng, Khách hàng, and Đăng xuất. The main area has a header "DE LUNA HOTEL" and a search bar. Below the search bar is a button "Thêm khách hàng". The main content is a table titled "All (56)" showing a list of 15 customers. The columns are: Mã khách hàng, Khách hàng, Username, Email, and Số điện thoại. Each row contains a checkbox, the customer's ID, name, username, email, and phone number. To the right of each phone number are two icons: a trash can and a pencil.

	Mã khách hàng	Khách hàng	Username	Email	Số điện thoại
<input type="checkbox"/>	AAAAAAI23	Nguyễn Thị A	Nguyễn Thị A	anguyenth@gmail.com	09799901233
<input type="checkbox"/>	AAAAAAI23	Lê Thị B	Lê Thị B	user123@gmail.com	09799901234
<input type="checkbox"/>	AAAAAAI23	Đặng Thị C	Đặng Thị C	example_mail2023@gmail.com	09799901235
<input type="checkbox"/>	AAAAAAI23	Trần Văn H	Trần Văn H	sample.email@gmail.com	09799901236
<input type="checkbox"/>	AAAAAAI23	Lê Thị Thu D	Lê Thị Thu D	anon.user0@gmail.com	09799901237
<input type="checkbox"/>	AAAAAAI23	Ngô Thị E	Ngô Thị E	mail.test2024@gmail.com	09799901238
<input type="checkbox"/>	AAAAAAI23	Lê Hoang D	Lê Hoang D	temporary.user@gmail.com	09799901239
<input type="checkbox"/>	AAAAAAI23	Lưu Thị F	Lưu Thị F	emailbox002@gmail.com	09799901240
<input type="checkbox"/>	AAAAAAI23	Hoàng Nguyễn Henry	Hoàng Nguyễn Henry	account.login123@gmail.com	09799901241
<input type="checkbox"/>	AAAAAAI23	Nguyễn Văn X	Nguyễn Văn X	no.reply123@gmail.com	09799901242
<input type="checkbox"/>	AAAAAAI23	Ngô Thanh P	Ngô Thanh P	no.reply129@gmail.com	09799901243
<input type="checkbox"/>	AAAAAAI23	Trần Minh H	Trần Minh H	randomemail2022@gmail.com	09799901244
<input type="checkbox"/>	AAAAAAI23	Lê Thị Thu Diễm	Lê Thị Thu Diễm	user120@gmail.com	09799901245

Figure 5. 20. Admin Customer Management Page Interface (Source: Authors)

The customer management page for the admin allows users to perform actions such as creating new customers, searching, sorting, editing and deleting customers. When the user selects the pencil icon, a modal popup will open, allowing the user to edit the customer's information. If the trash icon is clicked, the system will delete the customer from the database.



The screenshot shows a modal window titled "Sửa thông tin" (Edit information) over a list of phone numbers. The modal contains fields for Name (user3), Phone (098647512), Gender (dropdown menu), and Date of Birth (12/4/2017). On the right, there's a vertical list of phone numbers with edit and delete icons next to them. At the bottom of the modal, it says "1 - 5 of 10".

Số điện thoại
098647512
012312323
0123456789

Figure 5. 21. Edit Customer Information (Source: Authors)

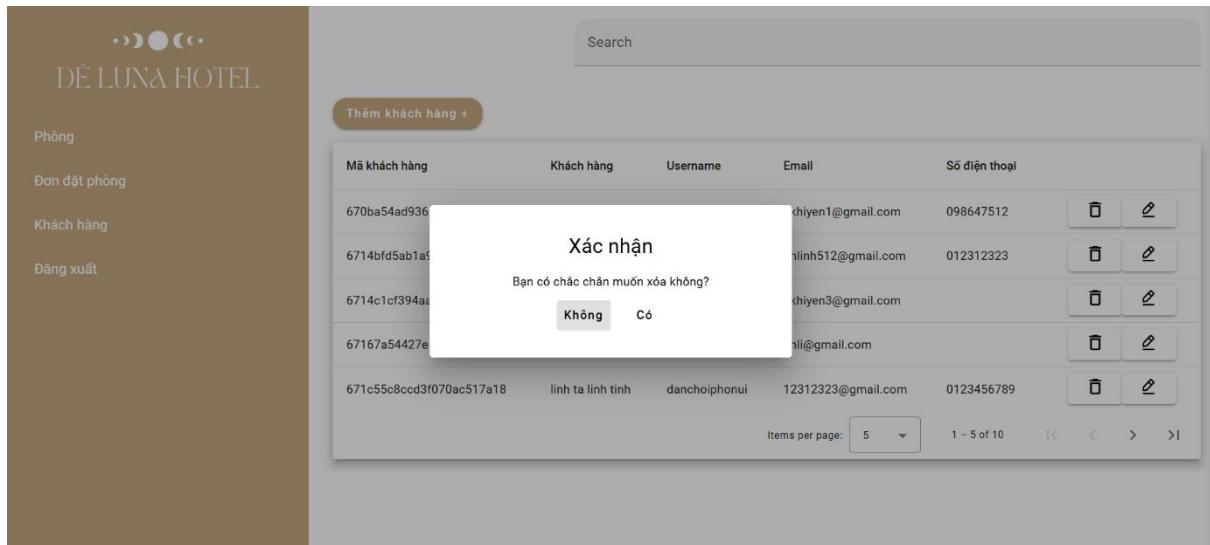


Figure 5. 22. Delete Customer Information (Source: Authors)

Room Management Page

The admin can create rooms, view a list of available rooms, edit information about each vacant room and delete rooms from the system as needed. This functionality helps streamline room management by providing quick access to room availability and the option to manage room inventory efficiently.

Danh sách phòng				
Thêm phòng +				
Mã phòng	Tên loại phòng	Giá	Tình trạng	
672264e941e9afb5d84056de	Deluxe Hollywood	3240000	Đang khả dụng	
6722652441e9afb5d84056e2	Deluxe Twins	3140000	Đang khả dụng	
6722653241e9afb5d84056e6	Deluxe Twins	3140000	Đang khả dụng	
6722655641e9afb5d84056ea	Deluxe King	3140000	Đang khả dụng	
6722656041e9afb5d84056ee	Deluxe King	3140000	Đang khả dụng	
6722658841e9afb5d84056f2	Executive King	3740000	Đang khả dụng	
672267af41e9afb5d84056f6	Executive King	3740000	Đang khả dụng	
672267d341e9afb5d84056fa	Executive Twins	3740000	Đang khả dụng	
672267e141e9afb5d84056fe	Executive Twins	3740000	Đang khả dụng	
6722680541e9afb5d8405702	Executive Hollywoods	3840000	Đang khả dụng	

Figure 5. 23. View Available Rooms List (Source: Authors)

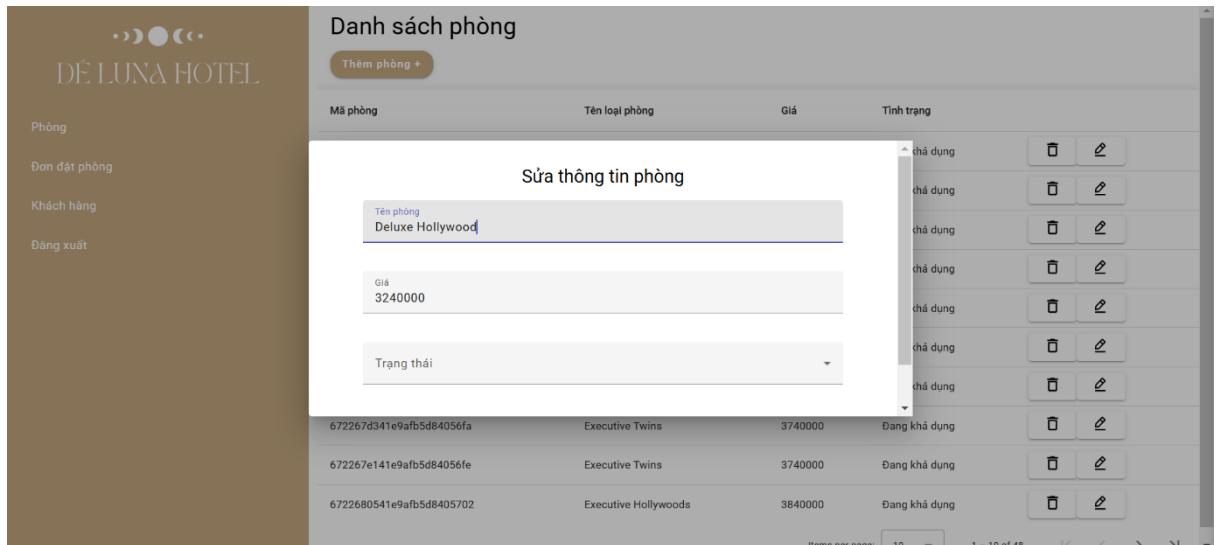


Figure 5. 24. Edit Room Details (Source: Authors)

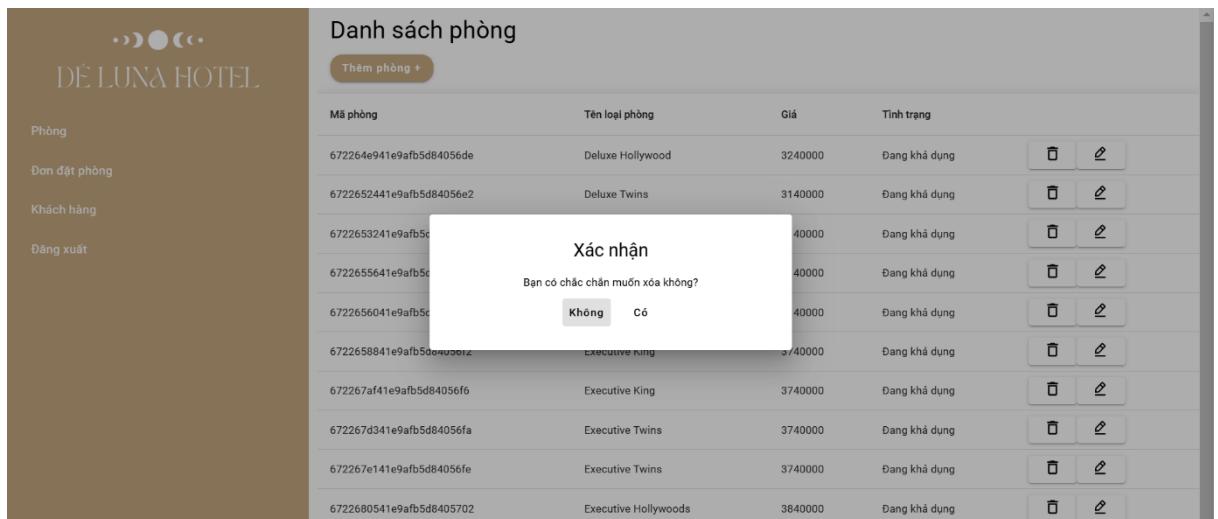
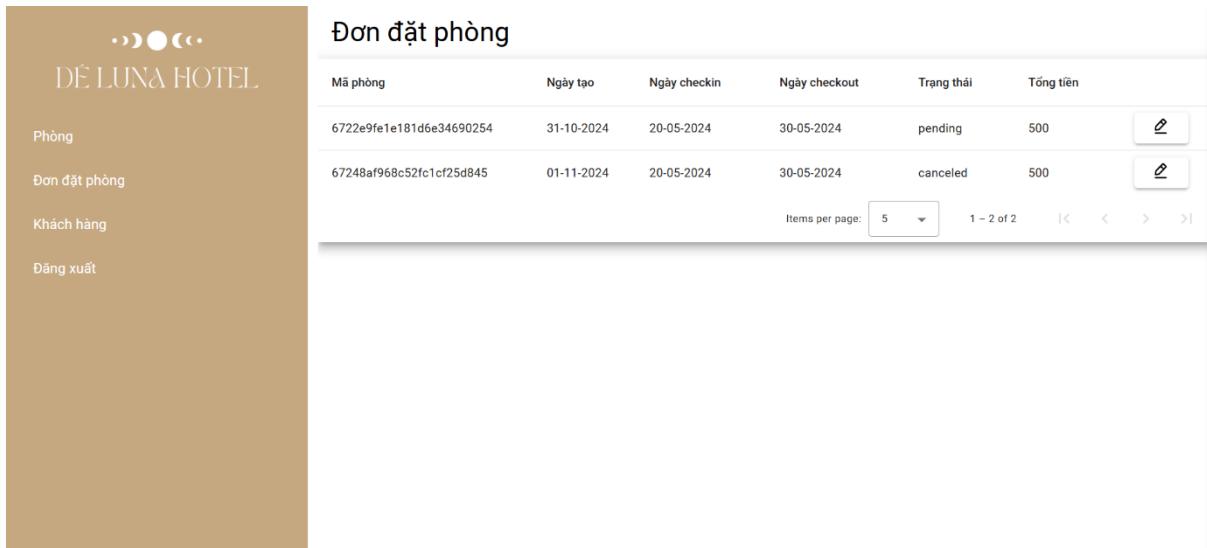


Figure 5. 25. Delete Room (Source: Authors)

Booking Details Page

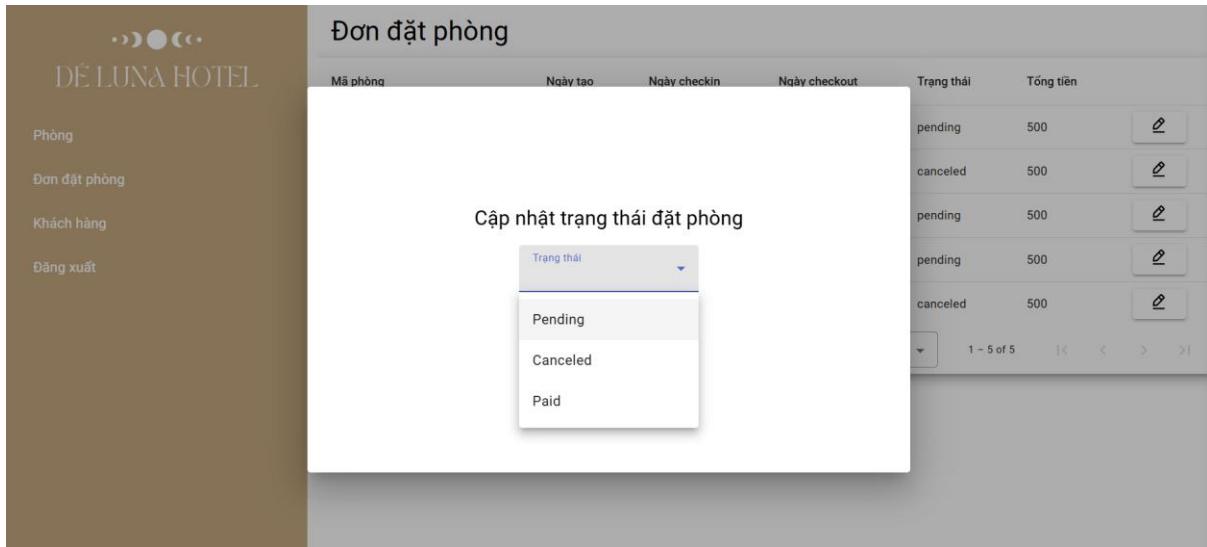
The admin page provides the necessary functions to manage hotel rooms. Admins can view a list of all bookings and update status booking (pending, cancelled or done).



Đơn đặt phòng						
Mã phòng	Ngày tạo	Ngày checkin	Ngày checkout	Trạng thái	Tổng tiền	
6722e9fe1e181d6e34690254	31-10-2024	20-05-2024	30-05-2024	pending	500	
67248af968c52fc1cf25d845	01-11-2024	20-05-2024	30-05-2024	canceled	500	

Items per page: 5 | < < > >|

Figure 5. 26. Booking Details (Source: Authors)



The screenshot shows a modal window titled "Cập nhật trạng thái đặt phòng" (Update booking status) over a list of bookings. The modal has a dropdown menu labeled "Trạng thái" (Status) with three options: "Pending", "Canceled", and "Paid".

Mã phòng	Ngày tạo	Ngày checkin	Ngày checkout	Trạng thái	Tổng tiền	
				pending	500	
				canceled	500	
				pending	500	
				pending	500	

1 – 5 of 5 | < < > >|

Figure 5. 27. Edit Status Booking (Source: Authors)

Page Logout

Log Out Page for Admin: Confirm your decision to log out of the admin panel to securely end your session. Click “Log Out” to proceed or “Cancel” to remain logged in.

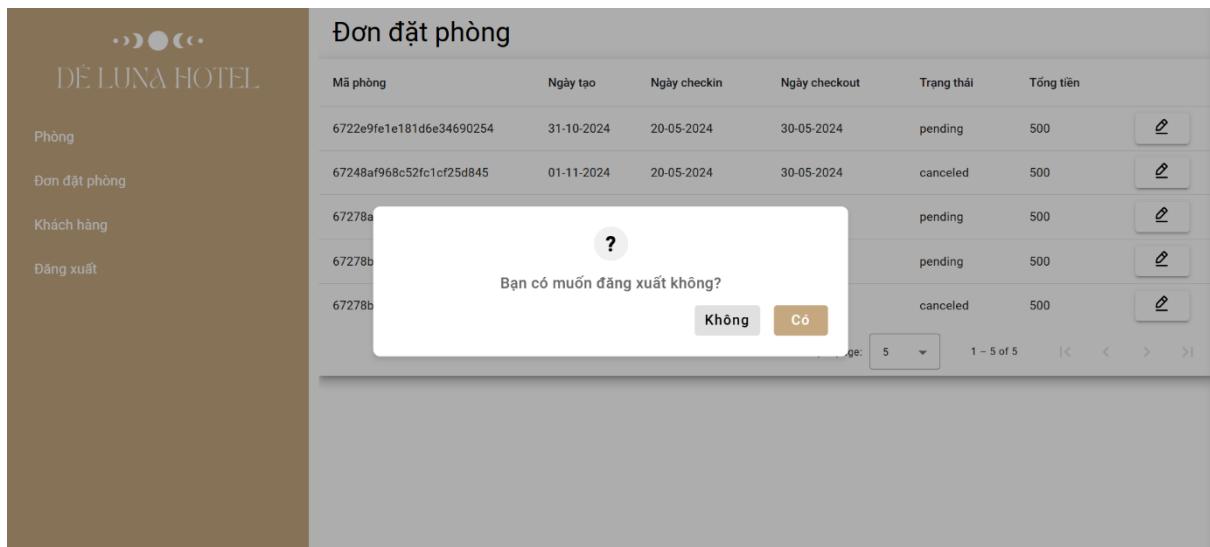


Figure 5. 28. Admin logout interface (Source: Authors)

5.2. Validation

Validation is data validation rules, improving input quality by checking data and rejecting any entries that do not meet the set conditions. This helps reduce data entry errors and improve data quality (Tilley, S., & Rosenblatt, H. J., 2017).

5.2.1. User input

Table 5. 1. Validation for user input (Source: Authors)

Data Name	Type	Note	Warning
Full name	String	Required	Cannot be left blank
Username	String	No spaces, no punctuation	Alert if conversion fails
Password	String	Must include at least 8 characters, use at least 1 lowercase letters, 1 uppercase letters, 1 numbers and 1 special characters	Number of characters less than 8 Does not include uppercase letters, numbers, or special characters
Email	String	Must be in the correct email format (e.g., example@gmail.com)	Incorrect format (missing @)

Phone number	String	Phone number must include 10 digits (for Vietnam) No spaces, letters, or special characters	Character count greater than 10 Contains special letters and characters
Avatar	String	Only png, jpg, jpeg formats are accepted	Select images in other formats
Date of birth	Date	The date of birth must not be greater than the current date	The date of birth must not be greater than the current date
Check in date	Date	The selected date is greater than the current date	Must be greater than or equal to the current date
Check out date	Date	The date chosen is smaller than the current date	Must be greater than Check in date

5.2.2. Admin Input

Table 5. 2. Validation for admin input (Source: Authors)

Data Name	Type	Note	Warning
Room price	String	Must not be less than or equal to 0	Price less than or equal to 0
Room image	String	Only png, jpg, jpeg formats are accepted	Select images in other formats
Restaurant signature images	String	Only png, jpg, jpeg formats are accepted	Select images in other formats

5.3. Database design

5.3.1. Overview and purpose

The database design provides a structured foundation for managing data within the application. These are 5 collections include:

- Users Collection: To store user information including login credentials, contact details, handling password resets and account verification.
- Rooms Collection: Captures detailed room information such as room type, amenities, status, and pricing, supporting inventory tracking and availability checks for booking.
- Restaurants Collection: Contains information about restaurants provide useful information for customer.
- Bookings Collection: Manages booking details, linking users to reserved rooms with associated check-in/check-out dates, total amounts.
- Email Promotion Collection: Stores email subscriptions for promotional content to email marketing.

5.3.2. Deliverables

MongoDB is an open source software for managing NoSQL databases. The MongoDB environment provides us with a server that you can start and then create multiple databases on it using MongoDB. With NoSQL databases, data is stored in the form of collections and documents.

The way MongoDB databases store collections is similar to the way MySQL databases store tables. Inside the collection, we have documents. These documents contain the data that you want to store in the MongoDB database, and a collection can contain multiple documents. Also, because of its schema-less nature, one document does not have to be the same as another. Documents are created using fields. Fields are key-value pairs in a document, just like columns in a relational database. The values of the fields can be any BSON data type such as double, string, boolean, etc. MongoDB stores data in the BSON document format. Here, BSON represents the binary encoding format of JSON documents (the B in BSON stands for Binary). In other words, in the backend, the MongoDB server converts JSON data into a binary form, called BSON, and this BSON can be stored and queried more efficiently. The maximum size of a BSON document is 16 MB. In MongoDB documents, they are allowed to store nested data. This nesting of data allows you to create complex relationships between data and

store them in the same document, making the process of working and fetching data more efficient than SQL.

5.3.3. Identify collection, field, data types

Table 5. 3. Description of the database (Source: Authors)

Collection	Field	Description	Data Type	Require
Users	user_id	User ID	ObjectID	True
	username	User's login username	String	True
	password	Password	String	True
	fullname	User's full name	String	True
	email	User's email	String	True
	phone_number	Phone number	String	True
	avatar	Profile picture	String	False
	gender	Gender (Male/Female/Unknown) Default: Unknown	String	False
	dob	Date of birth	Date	False
	address	Address	String	False
	status	User login status (Login/Logout)	String	False
	verified	Email verified status (active/inactive)	String	False
	isAdmin	True/False (admin rights)	Boolean	False
	reset password token	Password reset token	String	False
	reset password expire	Password reset token expiration date	Date	False

	email_verification_token	Email verification token	String	False
	email_verification_expire	Email verification token expiration date	Date	False
	created_at	Account creation date	Timestamp	False
	updated_at	Date of information change	Timestamp	False
Rooms	room_id	Room ID	ObjectID	True
	room_number	Room number of room	String	True
	room_name	Room name	String	True
	room_price	Room price	Number	True
	room_type	Room type	String	True
	room_status	Room status (True: available, False: unavailable)	Boolean	True
	room_details.beds	Number of beds	String	True
	room_details.occupancy	Maximum occupancy	String	True
	room_details.size	Room size (e.g., 40m ²)	String	False
	room_details.bathroom	Bathroom type	String	False
	room_details.view	View (e.g., sea view)	String	False
	room_details.unique_feature	Unique features (e.g., Jacuzzi, Balcony)	String	False
	room_amenities	List of room amenities	Array of Strings	False
	room_img	Images describe the room	Array of	True

			Strings	
	created_at	Room creation date	Timestamp	
	updated_at	Date of information change	Timestamp	
Restaurant	restaurant_id	Restaurant ID	ObjectID	True
	restaurant_name	Restaurant name	String	True
	restaurant_detail.location	Restaurant location	String	True
	restaurant_detail.guest	Guest capacity	Number	True
	restaurant_detail.menu_detail	Menu details	String	True
	restaurant_detail.hours	Opening hours	String	True
	restaurant_menu	Restaurant menu (contains multiple dishes)	Array of Strings	True
	restaurant_signature.images	Signature dish images	String	True
	restaurant_signature.describe	Description of signature dishes	String	True
	created_at	Restaurant information creation date	Timestamp	False
Bookings	updated_at	Date of information update	Timestamp	False
	booking_id	Booking ID	ObjectID	True
	check_in_date	Check In date	Date	True
	check_out_date	Checkout date	Date	True

	contact_infor.full_name	Contact's full name	String	True
	contact_infor.gender	Contact's gender	String	True
	contact_infor.phone_number	Contact's phone number	String	True
	contact_infor.email	Contact's email	String	True
	contact_infor.address	Contact's address	String	True
	guest.name	Name of the guest	String	True
	checkin_time	Check In time (optional)	String	False
	special_request	Special request (optional)	String	False
	booking_status	Booking status (done: paid, pending: waiting for payment, cancel: after 15 mins or error)	String	True
	total_amount	Total bill amount	Number	True
	room_id	Room ID	String	True
	user_id	User ID	String	True
	created_at	Booking creation date	Timestamp	False
	updated_at	Date of information update	Timestamp	False
EmailPromotion	email_id	Email ID	ObjectID	True
	email	Email promotion	String	True
	created_at	Email subscribe creation date	Timestamp	False

CHAPTER 6: SYSTEMS IMPLEMENTATION

6.1. Coding

6.1.1 Search available room API

Table 6. 1. Coding Search available room description (Source: Authors)

Component	Description	Example/Action
Input Data (req.body)	Data sent by the user to the API, including check-in and check-out dates.	{ "check_in_date": "2024-11-20", "check_out_date": "2024-11-25" }
convertDate	A function that converts dates from the user's string format to the Date type.	Input: "2024-11-20" → Output: ("2024-11-20T00:00:00Z")
Date logic validation	Verifies that the check-in date must be earlier than the check-out date.	If checkInDate >= checkOutDate → return HTTP 400 error: "Check-in date must be before check-out."
Room.find()	Retrieves all active rooms (room_status: true) from the MongoDB database.	js Room.find({ room_status: true });
checkRoomAvailability	Checks each room to determine if it is available between the specified check-in and check-out dates.	Input: room.id, checkInDate, checkOutDate → Output: true or false

Available rooms list	An array containing rooms that are available after the availability check.	<pre>data": [{ "_id": "64f891adf5b0c6b5a1d1c123", "name": "Deluxe Room", "price": 100 }]</pre>
Room.findById()	Queries the details of each room in the available rooms list while excluding unnecessary fields.	<pre>js Room.findById(_id).select('-__v -createdAt -updatedAt');</pre>
Response (Success)	If rooms are available, returns available: true status along with the list of available rooms.	<pre>{ "available": true, "data": [{ "_id": "64f891adf5b0c6b5a1d1c123", "name": "Deluxe Room", "price": 100 }] }</pre>
Response (Error)	If no rooms are available, returns available: false status along with the reason.	<pre>{ "available": false, "message": "No rooms available for the selected dates." }</pre>
Error handling (catch)	When an error occurs during execution,	<pre>{ "status": 500,</pre>

	returns a server-side error.	"message": "SERVER SIDE ERROR" }
--	------------------------------	-------------------------------------

```
const express = require('express');
const router = express.Router();
const { createBooking, getAdminBookings, getUserBookings, cancelSelfBookingOrder } = require('../controller/bookings')
const { isAuthenticatedUser } = require('../middleware/verify_token');
const { verifyAdmin } = require('../middleware/verify_admin');
const { isRoomAvailable, searchRoomAvailable } = require('../middleware/isRoomAvailable');

// Test isRoomAvailable

router.route('/create-booking').post(isAuthenticatedUser, isRoomAvailable, createBooking);

router.route('/usergetbooking').get(isAuthenticatedUser, getUserBookings)

router.route('/admingetbooking').get(isAuthenticatedUser, verifyAdmin, getAdminBookings)

router.route('/search-room').post(searchRoomAvailable)

router.route('/cancel-booking/:id').patch(isAuthenticatedUser, cancelSelfBookingOrder);

module.exports = router;
```

Figure 6. 1. Router Search available room (Source: Authors)

```

exports.searchRoomAvailable = async (req, res, next) => {
  try {
    const { check_in_date, check_out_date } = req.body;
    const checkInDate = convertDate(check_in_date);
    const checkOutDate = convertDate(check_out_date);

    if (checkInDate >= checkOutDate) {
      return res.status(400).json(createError(400, 'Ngày check-in phải trước ngày check-out.'));
    }

    const allRooms = await Room.find({ room_status: true });
    const availableRooms = [];

    for (const room of allRooms) {
      const isAvailable = await checkRoomAvailability(room._id, checkInDate, checkOutDate);
      if (isAvailable) {
        availableRooms.push(room);
      }
    }

    if (availableRooms.length === 0) {
      return res.status(400).json({
        available: false,
        message: 'No rooms available for the selected dates.'
      });
    } else {
      const mappedData = [];

      for (const room of availableRooms) {
        const roomData = await Room.findById(room._id).select('-__v -createdAt -updatedAt');
        mappedData.push(roomData);
      }

      return res.status(200).json({
        available: true,
        data: mappedData
      });
    }
  } catch (error) {
    console.error(error);
    return res.status(500).json(createError(500, 'SERVER SIDE ERROR'));
  }
};

```

Figure 6. 2. Controller searchRoomAvaiable function (Source: Authors)

6.1.2 Verify admin

Table 6. 2. Verify admin function description (Source: Authors)

Component	Description	Example/Behavior
Purpose	Middleware to verify if the current user has administrative privileges.	Allows access to protected routes for admins; denies access to non-admins.

Input (req.user)	The user object is extracted from the request. It typically contains user details, including roles.	{ _id: " 67167a54427e564016e5fb1b"}
User existence check	Checks whether the user object exists in the request.	If req.user is null or undefined, returns a 404 Not Found response.
Admin privileges check	Verifies the isAdmin field in the user object.	If user.isAdmin is true, proceeds to the next middleware (next()).
Response for non-admins	Returns a 403 Forbidden response if the user is not an admin.	Response: { status: 403, message: 'Access denied. Only admin can access.' }
Error handling	Catches any unexpected errors during execution.	Logs the error and returns a 500 Internal Server Error response: { message: 'SERVER SIDE ERROR' }.

```

backend > src > middleware > js verify_admin.js > ...
1 // Middleware to check if user is an admin
2 exports.verifyAdmin = async (req, res, next) => {
3   try {
4     // Retrieve the user from the request object
5     const { user } = req;
6
7     // Check if user exists
8     if (!user) {
9       return res.status(404).json({ message: 'Sorry, User does not exist' });
10    }
11
12    // Check if user has admin privileges
13    if (user.isAdmin) {
14      return next(); // Proceed if the user is an admin
15    } else {
16      return res.status(403).json({ message: 'Access denied. Only admin can access.' });
17    }
18  } catch (error) {
19    console.error(error);
20    return res.status(500).json({ message: 'SERVER SIDE ERROR' });
21  }
22 };
23

```

Figure 6. 3. Middleware verify admin (Source: Authors)

6.2. Programming languages and tools

6.2.1. The Choice of the Language

Table 6. 3. Language used (Source: Authors)

Components	Tools used
Web browser	Chrome
Software coding	Visual Studio Code
Database	MongoDB
Programming language	JavaScript, HTML, TypeScript
Environment	NodeJS

6.2.2. Programming Tools

6.2.2.1. Angular

Angular is a JavaScript framework developed using TypeScript. Google created this framework with the goal of building web interfaces (Front-end) with a "minimal effort" approach. In addition to offering the benefits of a typical framework, Angular maintains a structure similar to that of a standard programming language, making it easier for developers to scale and maintain projects efficiently.



Figure 6. 4. Logo Angular (Source: Internet)

Advantages:



Figure 6. 5. Advantages of Angular (Source: Internet)

Custom Components: Angular allows users to create their own components, which can encapsulate functionality along with the logic for those components, resulting in

reusable parts. These components also integrate seamlessly with other components within the web application, supporting the development of a flexible and consistent user interface.

Data Binding: Data Binding is a technique that establishes a connection between the user interface (UI) and data through business logic. Angular enables the seamless movement of data from JavaScript code to the view without requiring manual coding. This ensures that the UI automatically updates to reflect changes in data, enhancing the user experience.

Dependency Injection: Angular supports Dependency Injection, allowing users to easily inject services and dependencies wherever needed in the application. This not only improves the testability of the code but also enhances the reusability of components, minimizing unnecessary duplication.

Testing: Testing is a crucial aspect of Angular from its first version (Angular 1). Angular provides the capability to test every component of your application, offering a significant advantage over other JavaScript frameworks. Automated testing helps identify issues early and increases the stability of the application.

Comprehensive: Angular is a comprehensive framework backed by Google, offering a full suite of utilities for modern web application development, particularly in server communication, data management, and security. This provides peace of mind and high performance for developers.

Browser Compatibility: Angular is cross-platform and compatible with various browsers such as Chrome, Firefox, and operating systems like Windows, macOS, and Linux. This ensures that applications developed with Angular can run reliably across all environments that users may utilize.

Disadvantages:



Figure

Figure 6. 6. Disadvantages of Angular (Source: Internet)

Steep Learning Curve: Angular has a steep learning curve due to its comprehensive framework, requiring users to understand several core concepts such as directives, modules, decorators, components, services, dependency injection, pipes, and templates. Advanced topics like AoT compilation and Rx.js make it even more challenging for beginners, making Angular 4 difficult to grasp initially.

Limited SEO Options: Angular offers limited SEO capabilities, with poor accessibility for search engine crawlers, which can make it difficult to optimize for search engines.

Verbose/Complex: A common complaint within the Angular community is the verbosity of the framework. It tends to be more complex and cumbersome to work with compared to other front-end tools.

Complex Directives: Angular's use of directives can be challenging to learn and implement effectively, especially for those new to the framework, adding to the complexity of development.

6.2.2.2. Mongo DB



Figure 6. 7. Logo Mongodb (Source: Internet)

MongoDB is an open source software for managing NoSQL databases. The MongoDB environment provides us with a server that you can start and then create multiple databases on it using MongoDB. With NoSQL databases, data is stored in the form of collections and documents.

The way MongoDB databases store collections is similar to the way MySQL databases store tables. Inside the collection, we have documents. These documents contain the data that you want to store in the MongoDB database, and a collection can contain multiple documents. Also, because of its schema-less nature, one document does not have to be the same as another. Documents are created using fields. Fields are key-value pairs in a document, just like columns in a relational database. The values of the fields can be any BSON data type such as double, string, boolean, etc. MongoDB stores data in the BSON document format. Here, BSON represents the binary encoding format of JSON documents (the B in BSON stands for Binary). In other words, in the backend, the MongoDB server converts JSON data into a binary form, called BSON, and this BSON can be stored and queried more efficiently. The maximum size of a BSON document is 16 MB. In MongoDB documents, they are allowed to store nested data. This nesting of data allows you to create complex relationships between data and store them in the same document, making the process of working and fetching data more efficient than SQL.

Advantages:

No schema: Like other NoSQL databases, MongoDB does not require predefined schemas.

MongoDB stores any type of data: This gives users the flexibility to create as many fields in a document as they need, and makes scaling a MongoDB database easier than with a traditional relational database.

Document-oriented: One of the advantages of using documents is that these objects map to native data types in a number of programming languages. Having embedded documents also reduces the need for database connections, which can reduce costs.

Scalability: MongoDB's horizontally scalable architecture helps you create an application that can handle spikes in traffic as your business grows. Additionally, sharding allows the database to distribute data across a cluster of machines. MongoDB also supports creating data regions based on shard keys.

Third-party support: MongoDB supports a number of storage engines and provides a pluggable storage engine API that allows third parties to develop their own data storage engines.

Flexible large file storage: MongoDB has developed its own GridFS file system, which is similar to the Hadoop distributed file system. The use of the file system is intended to store files that exceed the size limit of BSON (16 MB per document).

Disadvantages:

Continuity: With automatic failover, users can only set up one master node in a MongoDB cluster. If the master node fails, another node automatically switches over to become the new master. This transition ensures continuity, but it is not instantaneous and can take up to a minute.

Write Limits: MongoDB's single master node also limits the speed at which data can be written to the database. Data writes must be written to the master node, and new information written to the database is limited by the capacity of that master node.

Data Consistency: MongoDB does not provide full referential integrity through the use of foreign-key constraints, which can affect data consistency.

Security: User authentication is not enabled by default in MongoDB databases. To secure your system against hacker attacks, you can manually set settings that block unfamiliar and unsafe connections.

6.2.2.3. NodeJS



Figure 6. 8. Logo NodeJS (Source: Internet)

NodeJS is designed to build highly scalable network applications and is an open-source, cross-platform runtime environment for JavaScript. NodeJS can run on a variety of operating system platforms such as Windows, Linux, and OS X, which are used to build web applications, mobile applications, or authentication applications,... Besides, NodeJS provides rich libraries in the form of various Javascript Modules. Thanks to its non-blocking processing capabilities, NodeJS can scale to manage millions of connections. Through the use of an event model and a single event loop that handles multiple connections at the same time.

Advantages:

NodeJS allows for the processing of multiple simultaneous connections without blocking. While other languages often block the process until the remote server responds, NodeJS is capable of handling multiple requests at once and acts as a client with third-party services.

Disadvantages:

The rapid development of code in NodeJS makes debugging more difficult. Also, the major drawback now is the lack of an out-of-the-box hosting environment for

NodeJS. Since NodeJS uses a single-threaded model, when the CPU is occupied, other requests are paused. The complex topics of the JavaScript language, such as archetypal inheritance, anonymous functions, and callbacks, make it difficult to learn, so learners often must master a language that is easier than before (Shah, 2017).

6.3. Testing

6.3.1. Overview

Objective: Perform Unit Testing, Integration Testing, and System Testing to identify errors occurring within individual components, during the integration of components, and throughout the entire system. Ensure that any errors are detected and resolved before the system is deployed.

Scope: Test all existing functionalities of the hotel booking website and the admin's hotel management page.

Testing Steps: Utilize manual testing methods and follow these steps:

- Step 1: Prepare test data.
- Step 2: Execute the test scenarios.
- Step 3: Collect and document the test results.

6.3.2. Test Scenarios

Table 6. 4. Scenarios for each type of testing (Source: Authors)

Unit Testing	Integration Testing	System Testing
---------------------	----------------------------	-----------------------

Scenario 1: User Registration	Scenario 6: Book a Room	Scenario 8: User Interface
Scenario 2: User Login		Scenario 9: Update User Information
Scenario 3: Calculate Room Price	Scenario 7: Manage Rooms	Scenario 10: Manage Bookings
Scenario 4: Check Room Availability		Scenario 11: Forgot Password
Scenario 5: User Management		

6.3.3. Test results

- Number of test scenarios executed: 11
- Number of detected issues: 12 issues
 - + 7 defects on the user website
 - + 5 defects on the admin website
- Test coverage: 100%
- The detected defects are detailed in the table below:

Table 6. 5. Issues discovered in the hotel's website (Source: Authors)

No.	Issues Name	Description	Priority	Solution
1	No notification when registering	<p>Steps: Register an account Access the website. Select “Create New Account”.</p> <p>Expected result: A notification is displayed if the registered username or email already exists.</p> <p>Actual result: The system does not notify the user.</p>	Medium	Add a pop-up notification to inform the user that the username/email already exists.
2	No notification	Steps: Log in using a successfully created account.	Medium	Add a pop-up notification to inform the user

	for incorrect login details	Expected result: A notification is displayed if the username or password is incorrect. Actual result: The system does not notify the user.		of incorrect login details.
3	Unable to save new booking to the database	Steps: Book a room. Check room availability. Select the desired room. Fill in the information. Expected result: After submitting the filled information, the booking is created and saved to the database. Actual result: It is not saved to the database.	High	Check again input user filled and input of API Create Booking (missing, wrong data type, extra) and then call API again from backend.
4	Fails to catch incorrect data input	Enter date of birth. Expected result: Future dates cannot be selected. Actual result: Users can enter any date.	High	Add function reject user selecting future dates.
5	Users cannot update their own information	Steps: Change account information. Select to edit information. Fill in the required information. Expected result: Updated information is saved to the database and displayed in the interface. Actual result: Changes are not saved or displayed.	High	Check again input user filled and input of API Update User (missing, wrong data type, extra) and then call API again from backend.

6	Unable to select the "Forgot Password" option	<p>Steps: Forgot Password. Select Login. Select Forgot Password.</p> <p>Expected result: The website redirects the user to the forgot password page, where they can enter their registered email to receive a password reset link.</p> <p>Actual result: The website redirects to the homepage.</p>	High	Hosting Website and Readjusting hyperlinks
---	---	---	------	--

Table 6. 6. Issues detected in Admin's hotel management page (Source: Authors)

No.	Issues Name	Description	Priority	Solution
1	Button add room not working	<p>Steps: Add room Log in to the management page with the admin account Select “Room”</p> <p>Select “Add room”</p> <p>Expected result: A pop-up appears for Admin to enter information for new rooms</p> <p>Actual result: Cannot select "add room" button on the interface</p>	High	Add pop-up to fill in information when creating a new room and calling API from Backend
2	Room information cannot be edited	<p>Steps: Update room Log in to the management page with the admin account Select “Room”</p> <p>Select edit information</p> <p>Expected result: The changed information will be displayed</p>	High	Check again input user filled and input of API Update Room (missing, wrong data type, extra) and then call

		<p>on the interface and updated to the database</p> <p>Actual result: Changed information is displayed on the interface, except for the status. And do not update the database</p>		API again from backend.
3	Booking information cannot be edited	<p>Steps: Change booking information</p> <p>Log in to the management page with the admin account</p> <p>Select “Booking form”</p> <p>Expected result: Room status is displayed on the interface, unchanged information is maintained and updated in the database</p> <p>Actual result: The changed information is displayed on the interface, but the remaining information disappears and is not saved to the database.</p>	High	Coding Update Booking API in backend and call it in front end
4	Incorrect display of objects that need to be managed	<p>Steps: View customer list</p> <p>Log in to the management page with the admin account</p> <p>Select “Customers”</p> <p>Expected result: Show only non-admin users</p> <p>Actual result: Displays all users including admin</p>	Medium	Add coding filter only user in API Get User List Backend
5	Customer information	Steps: Edit customer information	High	Check again input user filled

	cannot be edited	<p>Log in to the management page with the admin account</p> <p>Select “Customers”</p> <p>Select Edit information</p> <p>Expected result: Changed information will be displayed on the interface and updated to the database</p> <p>Actual result: The changed information is displayed on the interface but not updated in the database</p>		and input of API Admin Update Use (missing, wrong data type, extra) and then call API again from backend.
--	------------------	---	--	---

6.4. Deployment

6.4.1. Steps in the deployment process

Step 1: Release

- Purpose: Prepare software and related resources for transfer to customers and deployment in production environments.
- Check system requirements: Determine the hardware and software requirements of the system to run the website (server configuration, operating system, database, etc.).
- Create release documents: Include version information, new features, bug fixes, required system configuration, tools and installation instructions.
- Make a copy of the source code: Ensure that all the source code and related resources such as images, CSS, JavaScript, etc. has been fully inspected and packaged.
- Environment configuration: Configure related services (database, web server, email system, etc.) and define parameters such as domain name, security settings and access rights.
- Compatibility testing: Ensure that the website is compatible with different browsers, mobile devices, and popular operating systems.

Step 2: Installation

- Purpose: Install the website on a production environment and ensure all components work correctly.
- Server and software installation: Install operating system, web server, PHP, MongoDB, and other necessary technologies.
- Transfer source code to server: Upload website source code to server via FTP, Git or automatic deployment tools.
- Server configuration and environment:
 - + Configure domain name, SSL (HTTPS), security rules (firewall, network security).
 - + Install and configure database (MongoDB).
 - + Configure other necessary services such as email (SMTP), integration with payment gateways, map services (Google Maps).
 - + Run the database installation script: Create database structure, import sample data, configure database tables.
 - + Run a test application: Check that the website can load and run basic functions correctly on a production environment (e.g. booking, payment).

Step 3: Updating

- Purpose: Update the website when there is a new version or error correction.
- Identify updated versions: Check and confirm new versions of your website, new features or security patches.
- Make a backup before updating: Back up all website data, including source code and database, to avoid data loss during the update process.
- Update source code:
 - + Upload source code updates to the server.
 - + If you use a version control system like Git, you can pull new updates from the source code repository.
 - + Ensure changes in source code do not disrupt website operations.
- Database updates: If there are changes in the database structure, database update scripts (migration) must be run and ensure data is not lost.

- Test after update: Make sure old features still work properly and new features have been deployed successfully.

Step 4: Adaptation

- Purpose: Adjust the website to suit the specific requirements of the customer or the operating environment at the customer site.
- Analyze local requirements: Learn about specific customer requirements or market-specific factors (e.g. local payment gateways, languages, currencies, special booking processes) .
- Configure specific features: Adjust website features to suit local market or requirements (e.g. add language options, adjust payment methods, integrate with other services) local service).
- Customize appearance and content: Adjust the user interface to suit the client's aesthetic and branding requirements (e.g. change logo, layout, colors).
- Improve performance and security: Improve website performance (optimize page load speed) and make security adjustments, including checking for security flaws and applying the latest patches.

6.4.2. Deployment diagram

Components:

- Internet: Users access websites from the internet
 - + Configuration optimized for performance
 - + Set up monitoring and alerts
 - + Prepare expansion plans when traffic increases
- Web Server: Handles HTTP requests from users
- Web App Server: Contains hotel booking web application
- Entity Server: Manages the main entities of the system, including:
 - + Reservation Management (replaces Patient Management)
 - + Room and Facilities Management (replaces Medical Inventory)
 - + Payment Management (replacing Medical Accounting)
 - + Persistence: Store data

- Database Server: Stores all system data (using Oracle)
- Client/Server App Server: Handles requests from customer applications
- Local Area Network: Internal network connects system components
- Workstation Clients: Workstations for staff to manage and operate the reservation system

Deployment diagram:

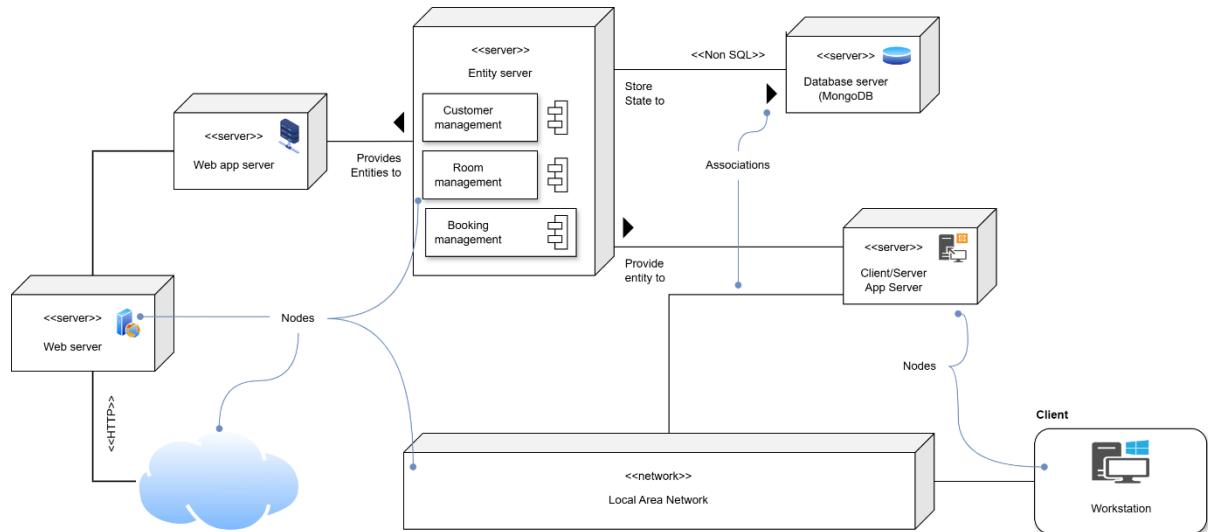


Figure 6. 9. Deployment diagram (Source: Authors)

6.5. User training

User training scenarios for customers and admins are described in the 2 tables below:

Table 6. 7. Customer training for hotel booking website Dé Luna Hotel (Source: Authors)

Who	The hotel booking website "Dé Luna Hotel" targets users from beginners to experienced customer. Therefore, it is necessary to create different user experiences to ensure ease of use for both novices and professionals.	
Where	Website for online hotel booking, which is published for users .	
Training Method	Content	Advantages

Intuitive User Interface	<ul style="list-style-type: none"> - User-friendly interface design - Search and filter tools for rooms - Simplified booking process 	<ul style="list-style-type: none"> - Easy to learn and use - Enhances user experience
Online Wizards	<ul style="list-style-type: none"> - Step-by-step booking guidance - Personal information input - Payment method selection 	<ul style="list-style-type: none"> - Reduces user errors - Increases booking completion rates
Online Help	<ul style="list-style-type: none"> - FAQ - Detailed user guides - Context-sensitive help 	<ul style="list-style-type: none"> - Quick answers to queries - Reduces support team workload
Tutorial Videos	<ul style="list-style-type: none"> - Advanced search feature usage - Group booking instructions - How to use discount codes 	<ul style="list-style-type: none"> - Visual and easy to understand - Can be rewatched multiple times
Live Webinars	<ul style="list-style-type: none"> - Introduction to new features - Money-saving booking tips - Live Q&A sessions 	<ul style="list-style-type: none"> - Direct interaction - Updates on latest information
Interactive User Guide	<ul style="list-style-type: none"> - Booking process simulation - Practice searching and comparing rooms - Managing bookings and cancellations 	<ul style="list-style-type: none"> - Learning through practice - Increases confidence in system use

Table 6. 8. Admin training for hotel booking management website Dé Luna Hotel

(Source: Authors)

Who	The hotel booking management website, 'Dé Luna Hotel,' targets users ranging from beginners to experienced admins.	
Where	Website for managing online hotel bookings, which is private and for internal admin use.	
Training Method	Content	Advantages
Intuitive User Interface	<ul style="list-style-type: none"> - Manage rooms: add, edit, delete rooms. - Manage booking status (done, 	<ul style="list-style-type: none"> - Easy to use with a user-friendly interface.

	<p>canceled, pending).</p> <ul style="list-style-type: none"> - Manage customer information. 	<ul style="list-style-type: none"> - Enhances user experience, reduces training time.
Online Wizards	<ul style="list-style-type: none"> - Step-by-step guidance for updating room details. - Update booking status. - Manage customer information. 	<ul style="list-style-type: none"> - Clear instructions help reduce user errors. - Increases accuracy in data updates.
Online Help	<ul style="list-style-type: none"> - Frequently asked questions about room and booking management. - Guides on updating and changing customer information. 	<ul style="list-style-type: none"> - Quick answers to common queries. - Reduces the workload for the support team.
Tutorial Videos	<ul style="list-style-type: none"> - Instructions on how to use room search and filter tools. - Managing confirmed bookings and cancellations. 	<ul style="list-style-type: none"> - Easy to understand and can be rewatched multiple times. - Suitable for learners at all levels.
Live Webinars	<ul style="list-style-type: none"> - Introduction to new system features. - Tips on cost-effective room and booking management. - Live Q&A sessions. 	<ul style="list-style-type: none"> - Direct interaction with experts. - Updates on the latest information and instant answers to questions.
Interactive User Guide	<ul style="list-style-type: none"> - Simulated guidance on room management and booking status. - Practice updating and managing customer information. 	<ul style="list-style-type: none"> - Hands-on learning increases confidence in using the system. - Enhances interactivity and long-term retention.

CHAPTER 7: SYSTEMS SECURITY AND SUPPORT

7.1. Threat Prediction and Analysis

Table 7. 1. Threat Prediction and Analysis (Source: Authors)

Threat	Category	Solution
Extortion	Hacker steals trade secrets and threatens to release them if not paid.	<ul style="list-style-type: none">- Data Encryption: Use strong encryption algorithms (like AES-256) to secure sensitive customer information, such as personal data and payment details.- Intrusion Detection System (IDS/IPS): Implement monitoring systems to detect and prevent unauthorized activities that could lead to extortion attempts.- Access Control Management: Limit access to critical information and use two-factor authentication (2FA) for administrator accounts.
Hardware and software failures	Router stops functioning, or software causes the application server to crash.	<ul style="list-style-type: none">- Backup and Redundancy: Deploy a redundant server infrastructure and perform regular data backups to secure locations (both on-premises and cloud).- Load Balancer: Use a load balancer to distribute traffic and ensure the system remains operational even if a server fails.- Routine Maintenance and Checks: Conduct periodic checks on hardware and software components to ensure they are in good working condition, minimizing technical failure risks.

Hardware and software failures	Employee accidentally deletes a file.	<ul style="list-style-type: none"> - Security Training: Train employees on security practices, including recognizing phishing emails, handling sensitive information, and protecting passwords. - Role-Based Access Control: Configure strict role-based access controls to ensure only necessary personnel can access sensitive data or critical functions. - Data Recovery Mechanism (Version Control): Implement a data backup and recovery system to allow for easy data restoration in case of mistakes.
Natural disasters	Flood destroys company building and networked systems.	<ul style="list-style-type: none"> - Disaster Recovery Plan (DRP): Develop and practice a disaster recovery plan, including data and system restoration procedures. - Multi-location Cloud Storage: Store data in multiple data centers across different geographical locations and use cloud storage for quick recovery if a disaster affects one area. - Invest in Disaster-Resistant Infrastructure: Ensure that devices, network systems, and infrastructure are capable of withstanding natural disasters (such as floods or earthquakes).
Service failure	Electricity is disrupted and brings the entire system down for hours.	<ul style="list-style-type: none"> - Power Backup: Install an Uninterruptible Power Supply (UPS) system for temporary power and backup generators for longer outages. - Network Redundancy: Ensure the system has redundant network connections from multiple providers to reduce the risk of network disconnection.

		<ul style="list-style-type: none"> - Cloud Services: Consider moving some or all of the system to the cloud to enhance fault tolerance and recovery capabilities during service failures.
Software attack	A group plants destructive software, a virus, or a worm into a company network.	<ul style="list-style-type: none"> - Web Application Firewall (WAF): Set up a WAF to block common attacks like SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). - Anti-malware Software: Install antivirus and anti-malware software and perform regular scans to detect malicious code. - Frequent Updates and Patches: Regularly update and patch the system to mitigate security vulnerabilities.
Technical obsolescence	Outdated software is slow, difficult to use, and vulnerable to attacks.	<ul style="list-style-type: none"> - Regular Software Updates: Track and periodically update operating systems, frameworks, and other software components. - Remove Deprecated Components: Replace or remove outdated components that are no longer supported by vendors. - Technology Risk Assessment: Conduct periodic assessments to identify risks associated with using outdated technology and create an upgrade plan.
Theft of physical or intellectual property	Physical server is stolen, intellectual property is stolen or used without permission; may be physical or electronic.	<ul style="list-style-type: none"> - Data Encryption and IP Protection: Use encryption to protect intellectual property and restrict access to it. - Strict Physical Access Control: Use physical access control measures like surveillance cameras, security personnel, and magnetic locks.

		<ul style="list-style-type: none"> - Multi-Factor Authentication (MFA): Apply MFA for administrator accounts and accounts with access to sensitive information.
Trespass and espionage	Employee enters unlocked server room and views the payroll data on a forbidden system.	<ul style="list-style-type: none"> - Physical and Virtual Access Restriction: Grant access only to necessary personnel and use two-factor authentication (2FA) for admin accounts. - Monitoring and Logging: Monitor and log all system access activities to detect unusual behavior and prevent espionage. - Intrusion Detection: Deploy intrusion detection systems to respond quickly to suspicious activity.
Vandalism	Attacker defaces Web site logo or destroys CEO's hard drive physically or electronically.	<ul style="list-style-type: none"> - Regular Data Backups: Regularly back up data and system configurations to facilitate easy recovery in the event of vandalism. - Data Tampering Prevention: Use technologies like WAF and systems to protect against data tampering or alteration. - Intrusion Detection System: Use intrusion detection systems to promptly detect and block vandalism attempts.

7.2. Update and Maintenance System

7.2.1. Maintenance process

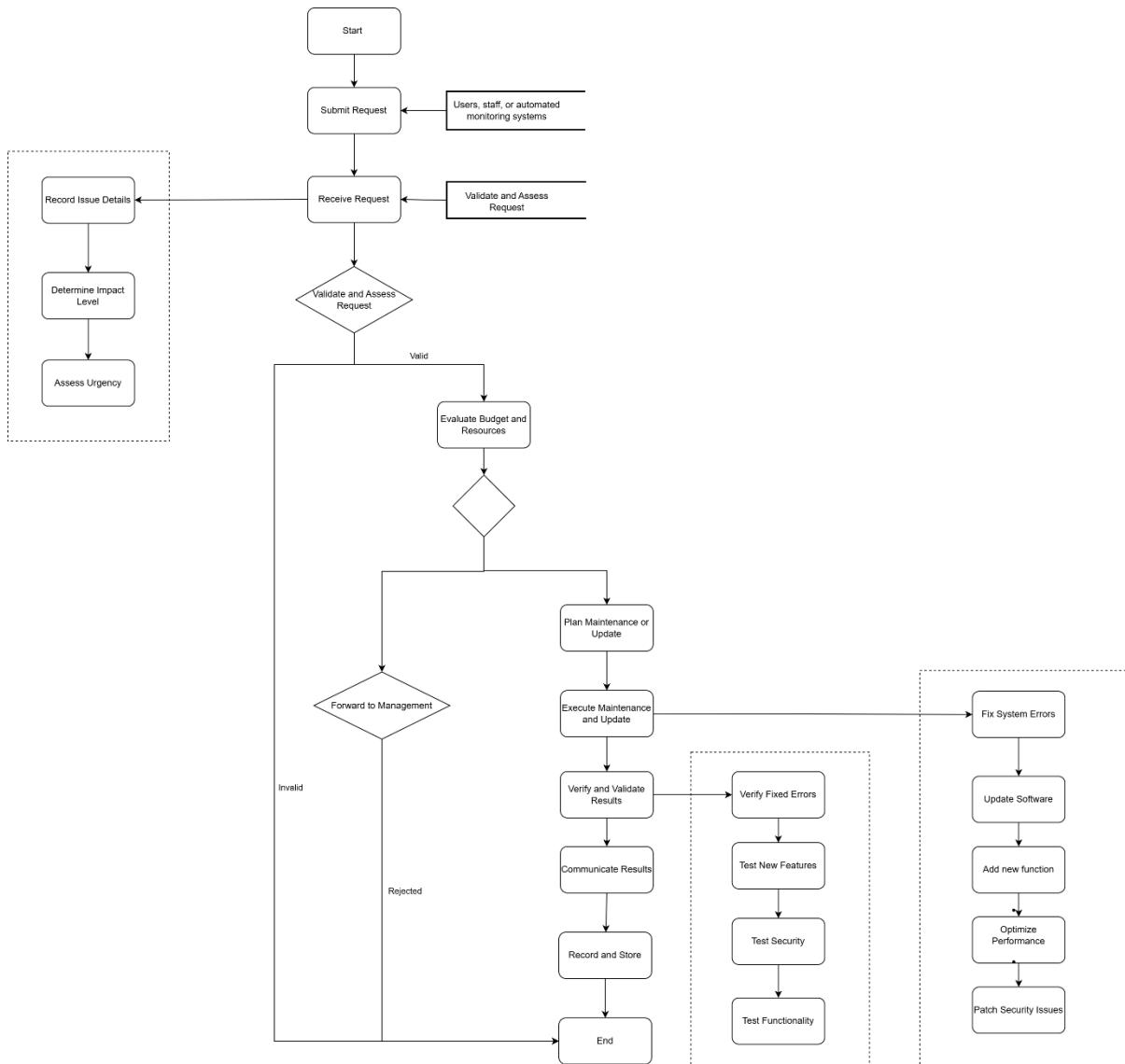


Figure 7. 1. Maintenance process (Source: Authors)

7.2.2. Update and Maintenance Plan

Purpose: The objective of this plan is to maintain the stability, security, and user satisfaction of the online hotel booking website by providing regular updates, bug fixes, security patches, performance improvements, and adding new features. The key goals include:

- Updating software and adding new features.
- Security and vulnerability fixes.
- Optimizing website performance.

- Improving user experience and adding new functionalities based on user feedback.

Detailed plan:

The detailed plan table described by the group is as shown in the table below:

Table 7. 2. Detailed plan (Source: Authors)

Plan	Content
Software updates and new features	<ul style="list-style-type: none"> Update Plugins and Modules: Check and update all plugins, extensions, or third-party services integrated into the website (such as payment gateways, booking engines, review systems, etc.). Add New Features: Implement new functionalities to improve user experience, such as AI-driven recommendations for hotel rooms, advanced filtering options, or improved booking flow. UI/UX Improvements: Update or redesign parts of the website's interface to improve usability, mobile compatibility, and user satisfaction, ensuring the website is always visually appealing and functional.
Security updates	<ul style="list-style-type: none"> Apply Security Patches: Regularly check and apply security patches for the operating system, web server, database, and other software components to avoid vulnerabilities. SSL Certificate Management: Ensure SSL certificates are valid and renewed before expiration, securing user data during transactions and interactions on the website. Perform Security Audits: Use tools like OWASP ZAP, Nessus, or other vulnerability scanners to identify potential security issues and patch them accordingly. Access Control Management: Review and update user permissions (e.g., admin, user, staff roles) to prevent unauthorized access to sensitive data and functionality.

Website Performance Optimization	<ul style="list-style-type: none"> - Check Page Load Speed: Use tools like Google PageSpeed Insights, GTmetrix, or Lighthouse to evaluate the website's performance and optimize factors that affect load time (e.g., caching, image optimization, minifying CSS/JS files). - Monitor Server Infrastructure: Ensure the server is capable of handling website traffic, preventing downtime and slow performance. Track resources such as memory, CPU, and bandwidth usage to detect potential issues. - Database Optimization: Regularly optimize database queries and perform database maintenance (e.g., clean up outdated data, fix performance issues) to ensure fast data retrieval.
Data Backup and Recovery	<ul style="list-style-type: none"> - Automated Backups: Set up regular backups for the entire website, including the database, configuration files, and any critical resources. - Test Backups: Perform regular tests to ensure the backups can be restored properly when needed. - Data Recovery Process: Ensure that a well-documented recovery process is in place and can be executed quickly in case of data loss or website issues.
Monitoring and User Support	<ul style="list-style-type: none"> - Monitor Website Uptime: Use uptime monitoring tools like UptimeRobot or Pingdom to ensure the website is online and available. Respond to downtime issues immediately. - User Support: Provide customer support channels (live chat, email) to assist users with booking, payment issues, or general inquiries related to the website. - Collect User Feedback: Continuously gather user feedback through surveys, ratings, or user behavior analytics to improve the website's functionality and overall experience.

Maintenance and Update Schedule:

The maintenance and update schedule described by the group is as shown in the table below:

Table 7. 3. Maintenance and update schedule (Source: Authors)

Schedule	Content
Daily	<ul style="list-style-type: none"> - Monitor website uptime and performance (e.g., availability, speed). - Address user issues and respond to support requests.
Weekly	<ul style="list-style-type: none"> - Perform data backups and verify backup integrity. - Optimize website performance (e.g., check for speed issues, optimize database). - Apply minor security patches if necessary.
Monthly	<ul style="list-style-type: none"> - Update plugins, modules, and software components. - Perform performance checks and identify areas for improvement - Analyze and implement user feedback for continuous improvement.
Quarterly	<ul style="list-style-type: none"> - Add new features or enhancements. - Review and implement security best practices (e.g., vulnerability scans, access control review). - Perform a comprehensive review of the backup and data recovery system.
Annually	<ul style="list-style-type: none"> - Update SSL certificates and ensure security compliance. - Review and update long-term performance optimization strategies. - Re-evaluate the design, UI/UX, and functionality to ensure the website remains competitive and user-friendly.

CHAPTER 8: PROJECT PLAN

8.1. Work Breakdown Structure

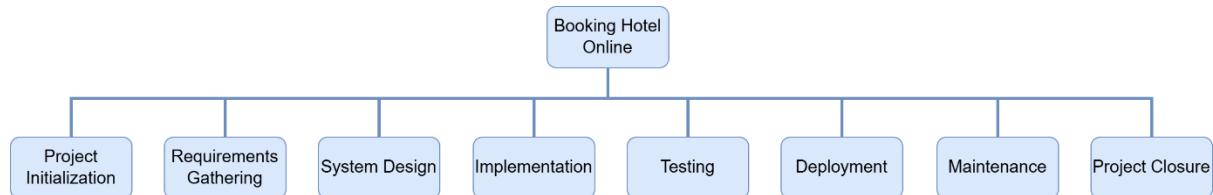


Figure 8. 1. Work Breakdown Structure (Source: Authors)

Project Initialization

- Define project objectives and goals for Dé Luna Hotel
- Conduct stakeholder analysis (e.g., hotel management, customers, and suppliers)
- Draft project charter and business case
- Set up project resources (team, budget, timeline)
- Establish the project schedule and milestones

Requirements Gathering

- Conduct interviews and surveys with stakeholders (management, customers, and staff)
- Collect functional requirements (booking, payment, customer management, etc.)
- Gather non-functional requirements (security, performance, scalability)
- Document customer expectations and operational needs
- Present findings to stakeholders and obtain sign-off

System Design

- Create system architecture for hotel management system
- Develop technical design documents for core functionalities
- Design Data Flow Diagrams (DFD) to model business processes
- Create Business Process Modeling Notation (BPMN) diagrams
- Design database structure and create a data dictionary (tables for rooms, reservations, payments, etc.)

- Design the user interface (website, mobile app, admin dashboard)

Implementation

- Implement hotel booking system (room reservations, availability, and pricing)
- Integrate customer management system (profile creation, booking history)
- Set up payment gateway for secure online transactions
- Integrate restaurant and event booking modules
- Implement loyalty programs and promotional offers
- Ensure multi-language and multi-currency support
- Develop back-end server and database structure
- Implement mobile responsiveness and compatibility across browsers
- Integrate admin panel for managing bookings, rooms, and customer data

Testing

- Conduct integration and system testing (ensure all modules work together)
- Perform user acceptance testing (UAT) with hotel staff and customers
- Compile testing reports and document issues
- Address and resolve identified issues (bugs, glitches, etc.)
- Conduct performance testing (load testing, scalability)
- Perform security testing (data encryption, vulnerability checks)

Deployment

- Prepare deployment plan (server setup, database configuration)
- Ensure compliance with regulatory standards (e.g., GDPR, PCI-DSS)
- Notify customers and staff about new system features
- Deploy the hotel booking system to production
- Train hotel staff on system operations (front desk, management)
- Set up monitoring tools for system performance

Maintenance

- Monitor system performance (uptime, load times, server health)
- Gather user feedback through surveys and reviews
- Conduct regular maintenance (bug fixes, updates)

- Address and resolve user-reported issues (customer complaints, technical problems)
- Plan for future feature expansions (additional services, integrations, upgrades)
- Implement a system update and maintenance plan
- Review and evaluate project outcomes against initial goals
- Document lessons learned and project successes
- Prepare final project report and handover to hotel management

8.2. Pert Chart

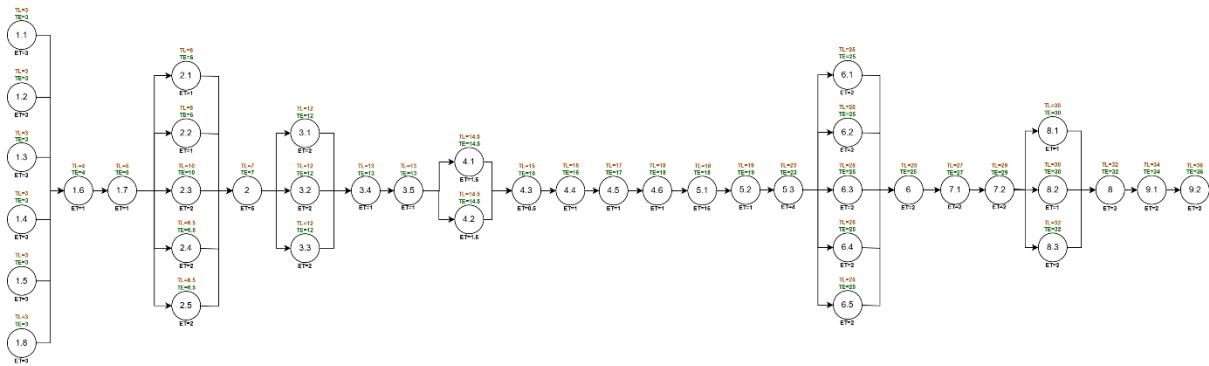


Figure 8. 2. Pert Chart (Source: Authors)

While performing this project, team divide the implementation process into 9 phases: (1. Project Overview, (2. Fact-finding and Requirements modeling, (3. Data and process modeling, (4. Object modeling, (5) User interface design and Data design, (6) Systems Implementation, (7) Systems Security and Support, (8) Project Plan and (9) Conclusion. The planning timeline is 36 days.

8.3. Gantt Chart

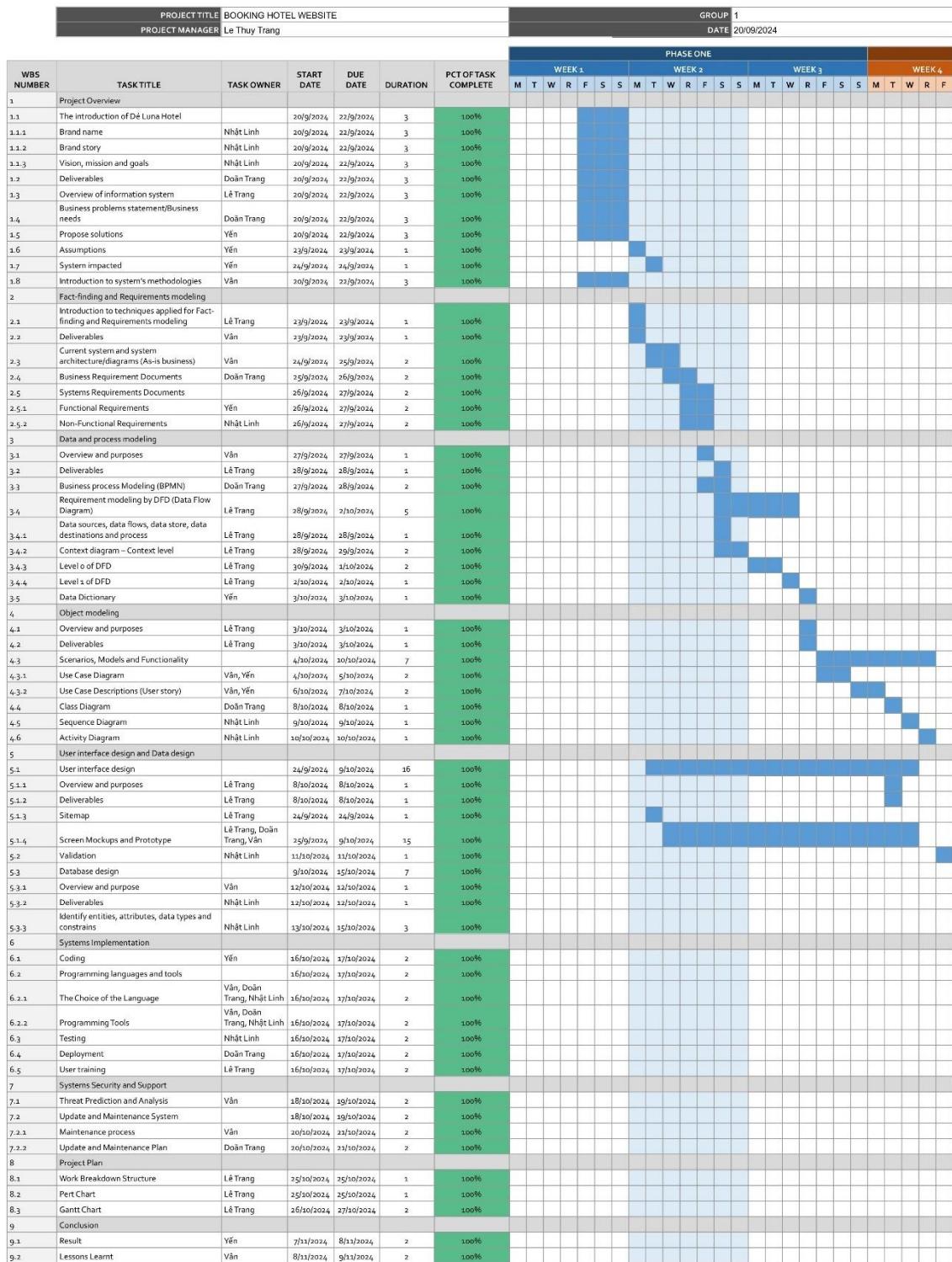


Figure 8. 3. Gantt Chart ([Link](#)) (Source: Authors)

The project finished in 36 days. All team members also successfully completed the assigned work, not late for submission.

CHAPTER 9: CONCLUSION

9.1. Result

The Booking Hotel System effectively manages hotel bookings, cancellations, and user information, providing a streamlined experience for both registered users and hotel administrators.

Table 9. 1: Result for Booking Hotel Website Project (Source: Authors)

No	Result	Description
1	Understanding of the Analysis and Design Process	Mastered the basic steps in the system analysis and design process, including requirement gathering, database design, and interface design
2	Optimized Business Processes	Modeling business processes by using BPMN, Use Case, Sequence, and Activity Diagrams, ensuring the website meets the hotel's requirements from booking to view details of rooms and services.
3	Efficient Data Management and Storage	Database design using MongoDB enables effective management of large data volumes, including user information, room details, restaurant details and booking transaction
4	Application of Knowledge to Real-World Situations	Applied theoretical knowledge to real-world scenarios through projects, improving problem-solving skills
5	Development of Teamwork Skills	Enhanced teamwork, communication, and collaboration skills to achieve common goals in the project
6	Assessment of the Impact of Information Systems	Identifying benefits and challenges in technology adoption

9.2. Lessons Learnt

Table 9. 2. Lessons Learned and Improvements for Booking Hotel Website Project (Source: Authors)

No	Issue/ What did not work well	What the team should have done instead?	Lessons learned
1	Lack of consulting with experts in the field led to several design and functionality decisions that were not fully optimized.	Seek feedback from experts in the field to gain insights into design and functionality decisions before finalizing them.	Consulting with experts provides valuable insights that can enhance product quality and align it better with industry standards.
2	No clear development roadmap for long-term expansion, resulting in challenges when trying to add features during development.	Create a roadmap with features that allow for easy scalability and expansion in future stages of development.	A clear roadmap helps the team plan effectively for future enhancements, minimizing issues when adding new features during development.
3	Lack of regular software updates led to certain modules becoming outdated and incompatible, negatively affecting system performance.	Regularly check for updates to keep the software current, ensuring compatibility and security.	Regular updates prevent outdated software issues, ensuring the system remains compatible and up to date with the latest standards.
4	No monitoring system was in place, making it difficult to identify and resolve issues during website operation.	Implement a monitoring system to quickly detect and resolve issues as they arise on the website.	A monitoring system provides real-time alerts, enabling the team to identify and resolve issues promptly, ensuring website stability.

5	Insufficient mobile optimization, making the website difficult to navigate and use on mobile devices.	Design with a mobile-first approach and conduct regular mobile usability tests throughout development.	Ensuring mobile optimization improves user experience on mobile devices, which is essential given the high rate of mobile users accessing the site.
6	Lack of accessibility testing made the website difficult for users with disabilities to navigate.	Apply accessibility principles to ensure the website is user-friendly for everyone, including those with hearing, visual impairments, and elderly users.	Ensuring accessibility expands the user base and complies with regulations on accessibility for people with disabilities.
7	Missing features for managing customer reviews and feedback limited the ability to improve services based on real feedback.	Provide features that allow users to leave ratings and feedback directly on the website to enhance services based on customer insights.	Listening to customer feedback helps improve service quality and increases user satisfaction.
8	Lack of methods for testing compatibility with multiple languages, making it challenging for international users.	Conduct language compatibility tests and optimize for different language regions.	Ensuring multilingual compatibility broadens the user base internationally and creates a better experience for global users.

Table 9. 3. Success Factors and Best Practices (Source: Authors)

No	What worked well	Lesson learned (Best Practice)
1	Analyzing competitor websites on the market helped the team visualize a website that meets the essential needs and features commonly desired by clients.	Researching competitor websites on the market helped the team identify the structure and key elements to focus on, as well as what features to prioritize and which ones to avoid for the specific client type. This is an important step in setting a clear goal and sustaining product development.
2	Maintaining weekly meetings helped the team communicate and quickly resolve issues. It also fostered connections and trust among team members.	Regular communication between team members helped everyone stay updated on each person's tasks and workload, making it easier to identify and address issues quickly. Establishing this communication routine is essential, especially for teams working in different locations, as it ensures everyone is aligned and prevents misunderstandings. Regular meetings also created a platform for team members to discuss challenges, seek solutions, and collaborate closely on project requirements. This facilitated efficient project development and encouraged prompt progress tracking.
3	Coordinating well among team members during the testing and completion phases helped create a final product that met quality	Close coordination between members, including regular updates on individual and team responsibilities, helped prevent redundant work and supported a smooth, on-schedule completion of the project.

	standards and performed as expected	This reinforced the importance of effective communication in achieving project goals. The final review by key team members also provided valuable feedback, ensuring the product met client requirements and maintained high quality and performance standards.
4	Assigning tasks reasonably and clearly to each team member helped balance responsibilities and promoted mutual support among members. This is an important skill for teamwork in any project.	Clearly assigning tasks helped each team member understand their role and responsibilities, fostering initiative and a proactive approach. When responsibilities are not clearly defined, it can lead to overlapping work and wasted effort. Properly distributing tasks also makes it easier to track progress and provide timely support when needed, thereby fostering stronger team collaboration. This is a key skill in teamwork, especially in complex projects with high workloads and deadlines. Clearly defined roles and responsibilities also boost team morale, helping each member focus on their specific tasks while achieving the overall project objectives
5	Adopting the Agile methodology brought flexibility and efficiency to the project. Dividing the work into short cycles (sprints) allowed for easy progress tracking and timely adjustments after each phase. After	Agile is an ideal methodology for projects that require frequent changes, as it provides a flexible approach to development. The team could track progress continuously through each sprint and make timely adjustments as

	<p>each sprint, the team could review completed parts, quickly address issues, and improve the product without disrupting the overall progress. This also allowed the team to experiment and modify features as needed, keeping the project aligned with initial goals and maintaining high quality.</p>	<p>needed. Agile not only optimized the workflow but also encouraged the team to focus on continuous improvements to meet project requirements accurately. This experience reinforced the value of Agile in delivering a high-quality product that could adapt quickly to evolving requirements, ensuring the project remained aligned with its initial goals and objectives.</p>
--	--	---

REFERENCES

- Bali, O. (2024). *Hotel Management Information System: What they are and why they matter*, Omni Hotelier. Available at: <https://omnihotelier.com/en/hotel-management-information-system-what-they-are-and-why-they-matter/> (Accessed: 05 November 2024).
- Figma. (n.d.). *Figma*. Retrieved October 5th 2024 from <https://www.figma.com/>
- IIBA, “A Guide to the Business Analysis Body of Knowledge (BABOK Guide) Version 3.0,” *International Institute of Business Analysis*, Ontario, 2015.
- Tilley, S., & Rosenblatt, H. J. (2017). *Systems analysis and design*. Cengage Learning.
- UKEssays. (November 2018). *Definition of fact-finding techniques information technology*. UKEssays. Retrieved October 5th 2024 from <https://www.ukessays.com/essays/information-technology/definition-of-fact-findingtechniques-information-technology-essay.php>
- Shah, H., & Soomro, T. R. (2017). *Node.js challenges in implementation*. Global Journal of Computer Science and Technology, 17(2), 73-83.