Technologies for Complex Web Based Platforms. Case Study: Hotel Booking and Resource Management Platform

Ștefan-Alexandru ȘINDILAR and Dana-Mihaela VÎLCU

***Abstract*—Resource management plays a vital role in the hospitality sector, requiring tools that optimize, automate, and simplify operations. This paper presents a web-based solution for hotel resource management, designed to deliver an efficient and user-friendly experience. The system features a React interface tailored to administrators, staff, and clients, enabling streamlined booking and management processes. The backend, developed with Node.js, integrates diverse services and external APIs to extend functionality, including Google account authentication, multi-factor authentication (MFA), and AI-driven tools. The OpenAI API generates automatic descriptions for rooms and hotels, while also powering a recommendation engine that provides personalized client suggestions. Online payments are facilitated through Stripe, supporting card transactions in RON and bank transfers in EUR, with automatic RON–EUR currency conversion. To ensure security and reliability, the platform includes email notifications and a validation mechanism verifying both company activity and responsible personnel. Identity validation is managed via a Python Flask server, employing OCR, facial recognition, and anti-spoofing techniques. Data management relies on MongoDB, ensuring scalability and flexibility for future platform growth. This solution demonstrates how modern web technologies and AI integration can enhance resource management in the hospitality industry.**

***Index Terms*—artificial intelligence, automation, facial verification, hospitality industry, software technologies, web platform.**

I. Introduction

In the current context of accelerated digitalization across key industries, the hospitality sector faces increasing challenges in optimizing resource management. Hotels are required not only to streamline internal operations but also to provide clients with a simplified, efficient, and enjoyable experience. Addressing these challenges demands the implementation of modern technological solutions that align the industry with contemporary digital practices, ensuring competitiveness and sustainability, as presented in [1].

This paper focuses on the development of a web-based platform dedicated to hotel resource management, designed to enhance automation, reduce human error, and save time in daily operations. The proposed solution addresses the needs of all stakeholders—clients booking accommodations, hotel staff managing resources, and administrators overseeing the platform—while also integrating auxiliary features such as real-time notifications, intuitive user interfaces, secure authentication, and identity verification. Through the use of modern web technologies, external APIs, and AI-driven functionalities, the platform aims to create a transparent, reliable, and scalable solution that redefines resource management in the hospitality industry.

Ș.-A. ȘINDILAR is with the Special Telecommunications Service, Bucharest, Romania (e-mail: sindilar.stefan@gmail.com).

D.-M. VÎLCU is with the Military Technical Academy “Ferdinand I”, Bucharest, Romania (e-mail: dana.vilcu @mta.ro).

II. Related Work

Understanding the current state of the hospitality industry is essential for developing an effective web-based platform for resource management. Modern hotels operate in a highly dynamic environment, where operational efficiency, service quality, and customer satisfaction are closely linked to the effective management of human, technological, and logistical resources. The sector has been rapidly adopting digital solutions to streamline operations, automate bookings, and optimize the use of resources, particularly in the post-pandemic context. Investments in IT infrastructure, integrated Property Management Systems (PMS), online booking platforms, and personalized digital services have become strategic priorities for hotels seeking to enhance competitiveness, customer loyalty, and operational sustainability.

The digital transformation of hotels relies on a combination of software solutions and platforms designed to manage resources efficiently and enhance customer experience:

- *Property Management Systems* platforms centralize hotel operations, including: front desk and room management, house keeping, customer data management, reporting and analytics, point of sale, back office, revenue management, and a connection with a central reservation system. These systems reduce human error, improve administrative efficiency, and allow hotels to manage resources in a structured and consistent manner. A complete characterization of PMS can be found in [2].
- *Online Travel Agencies* (OTA), such as Booking Holdings, Airbnb, or Expedia Group, act as intermediaries between clients and hotels, increasing visibility and occupancy without requiring significant proprietary infrastructure. By their own platforms OTAs provide hotels access to global markets and enhance competitiveness through extended digital presence [3].
- *Channel Managers* are these tools that synchronize availability, rates, and bookings between PMS and multiple OTAs in real time, preventing overbooking and enabling dynamic pricing strategies across all sales channels. They reduce manual errors and streamline operational management [4].

Digital Object Identifier 10.32754/JMT.2026.1.12

– *Central Reservation Systems* (CRS) are solutions used to consolidate reservations from multiple distribution channels, ensuring accurate and centralized management of availability and pricing [5]. *Booking Engines* integrated into hotel websites allow for direct reservations, reducing dependency on intermediaries and associated commissions [6].
– *Customer Relationship Management* (CRM) Systems support the personalization of guest experiences, collecting and analyzing customer data to enhance satisfaction and loyalty. They enable targeted marketing and proactive engagement with guests based on preferences and booking history [7].
– *Revenue Management Systems* (RMS) platforms optimize pricing and revenue by analyzing real-time data, market demand, and booking trends. They enable dynamic pricing strategies that maximize revenue per available room (RevPAR) and overall profitability [8].

Together, these technologies form a comprehensive ecosystem (Fig. 1) that supports automation, operational efficiency, revenue optimization, and personalized guest services. The integration of these systems ensures that hotels can manage resources effectively, respond to market demands dynamically, and maintain high service standards, which are crucial for long-term competitiveness and sustainability in the hospitality industry.

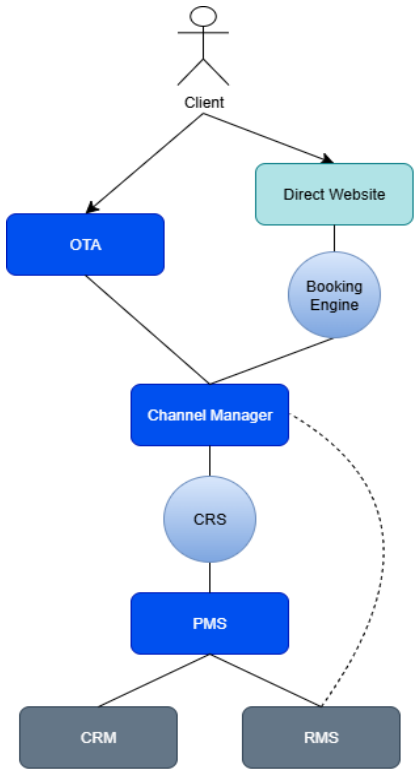

Figure 1. The technological ecosystem in the hospitality industry

Analyzing leading platforms in the hospitality industry provides valuable insights for the development of a dedicated web platform for hotel resource management. Examining their architecture, core functionalities, and integration with external systems allows the identification of best practices and technological strategies to guide platform design. The following paragraphs present the Booking.com platform.

A. A Brief Analysis of Booking.com

According to [9] and [10], Booking.com is one of the largest global Online Travel Agencies (OTAs), handling millions of users and daily transactions. The platform adopts a microservice architecture with asynchronous communication using an event-driven pattern to ensure system stability, scalability, and maintainability. The frontend focuses on delivering an intuitive user experience, while the backend comprises microservices responsible for resource management, reservations, payment processing, review management, and dynamic pricing.

Requests are routed through an API Gateway, which serves as a single entry point, handling authentication, authorization, and load balancing. Data management relies on a polyglot persistence model, using relational databases for transactional data and non-relational databases for unstructured or semi-structured data. Techniques such as data sharing, replication, and caching, including Content Delivery Networks (CDN), ensure high availability and fast response times. The platform also integrates third-party solutions, such as Elasticsearch[2] for efficient search and Kafka[3] for data streaming, to optimize performance. Advanced security measures, including data encryption, two-factor authentication (2FA), and continuous vulnerability monitoring, protect user information [11]. Moreover, Booking.com allows the integration of hospitality technology solutions to simplify property management and at the same time scale the business [12].

The following section presents the Airbnb platform, another major OTA, based on the information provided by [13] and other supplementary sources. Airbnb has evolved from a monolithic architecture (2008–2017) to a complex, scalable system based on microservices and macroservices. The migration involved extracting core components into independent services, enabling separate development, testing, and scaling. The current architecture includes services for user management, search, reservation processing, messaging, reviews, notifications, and dynamic pricing. An API Gateway manages request routing, authentication, and authorization.

B. A Brief History of Airbnb

Airbnb started (2008–2017) with a monolithic Ruby on Rails[4] architecture, which enabled a fast product launch but soon showed limitations in scalability, reliability, and feature development. To overcome these challenges, the platform migrated (2017–2020) to a microservices-based SOA (Service-Oriented Architecture), separating core components into independent services and ensuring stability through dual-read validation and shadow databases. This shift introduced complexity in managing services and databases, addressed by adopting Kubernetes[5], a Thrift-based communication framework[6], and the in-house Powergrid library for parallel task execution. Since 2020, Airbnb has been operating a hybrid model of microservices and macroservices, combining flexibility, scalability, and performance [14-15].

[2] https://www.elastic.co/elasticsearch
[3] https://kafka.apache.org/
[4] https://rubyonrails.org/
[5] https://kubernetes.io/
[6] https://thrift.apache.org/

 Digital Object Identifier 10.32754/JMT.2026.1.12

Airbnb allows the integration of services from various partners that provide hospitality-specific functionalities such as PMS and Channel Managers. These integrations facilitate process automation, synchronization of availability and pricing across multiple platforms, and provide centralized and automated control of operations [16].

Airbnb's backend combines Ruby on Rails and Node.js[7], with relational and non-relational databases, while the frontend uses React Native for cross-platform compatibility. Machine learning algorithms support personalized recommendations and dynamic pricing adjustments. External services such as Amazon Web Services (AWS)[8], Google Maps API[9], Firebase[10], Elasticsearch, and Braintree[11] enhance scalability, real-time communication, search efficiency, and payment processing. Security measures include two-factor authentication (2FA) and facial recognition for identity verification to prevent fraud and ensure trust within the community [17-18].

The analysis of Booking.com and Airbnb platforms provides a comprehensive and holistic perspective for the development of a web platform dedicated to the hospitality industry. By examining their architecture, components, and functionalities, a reference model and a starting point emerge for designing an efficient solution tailored to the requirements of resource management in the hospitality sector.

III. METHODS

Our proposed platform is designed to address the specific requirements of three main categories of users: the platform administrator, the hotel manager, and the client. Each category is provided with a dedicated interface that ensures an optimized interaction with the system according to the assigned responsibilities and level of access. The distinction between roles not only facilitates the streamlining of operations but also guarantees data protection and controlled access to resources.

From a functional perspective, the platform is structured hierarchically, with the administrator at the top, possessing a comprehensive view of all operations and resources. The administrator supervises the overall activity of the system, ensuring coherence, transparency, and compliance with platform objectives. The hotel manager operates at an intermediate level, managing a single hotel and the resources associated with it, including rooms, reservations, reviews, and revenue statistics. Finally, the client represents the end-user, whose interaction with the system is oriented toward exploring available accommodation, booking services, and providing feedback.

The modular design of the platform ensures that functionalities are specifically adapted to each type of user. In this way, the application achieves a balance between accessibility and security, allowing for an efficient distribution of tasks, a reduction of human errors, and an overall improvement of user experience across all interaction levels.

The platform provides intuitive, clear, and adaptive interfaces for each user type. Client interfaces should be responsive across devices, covering all functional use cases, while administrative and manager interfaces are optimized for larger screens, displaying comprehensive data through tables, charts, and complex editing components.

Security measures include robust authentication and authorization using JWT tokens with role-based access control, account activation, secure password management, and multi-factor authentication for privileged users. Data protection is ensured through HTTPS, encryption of sensitive information, and bcrypt hashing for passwords. XSS attacks are mitigated via content sanitization, and session management enforces token expiration and refresh mechanisms. Error messages avoid revealing sensitive information, while logging and auditing track abnormal behavior. Identity verification processes for managers prevent unauthorized access and ensure system integrity.

This section presents the overall architecture of the application (Fig. 2), highlighting the main components and the way in which they interact to provide a robust and efficient solution for resource management in the hospitality industry. The architecture is designed to optimize, simplify, and automate operations, while at the same time significantly improving both user efficiency and user experience.

The application is structured around three core components: the client application, the main server, and the identity verification server, each having well-defined roles to ensure the logical consistency and proper functioning of the platform. The client application enables users to interact with the system through an intuitive and personalized interface tailored to their role. It communicates with the other two components to ensure the delivery of all required functionalities.

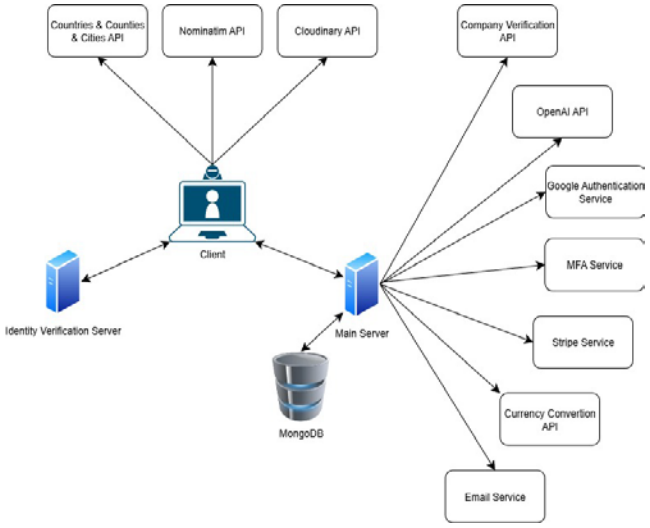

Figure 2. System architecture

The client application represents the user-facing component of the platform, designed to ensure intuitive and efficient interaction through a modern, component-based architecture. Developed with the React framework[12], it leverages reusable components, the Virtual DOM, and Single Page Application (SPA) principles [19] to provide scalability, fast rendering, and a seamless user experience.

[7] https://nodejs.org/en
[8] https://aws.amazon.com/
[9] https://developers.google.com/maps
[10] https://firebase.google.com/
[11] https://www.paypal.com/us/braintree
[12] https://react.dev/

The system is structured across several layers: a presentation layer (UI/UX) for visual interaction, Redux[13] for global state management, React Router[14] for dynamic and protected navigation, and RESTful API communication[15] (via Axios[16] and fetch[17]) with the main server, the identity verification server, and third-party services.

Styling is implemented using CSS, Bootstrap[18], and SCSS, which also enables a dark mode for administrative dashboards. User actions are supported by contextual feedback, while localStorage [20] ensures persistence of data during registration processes.

The client application communicates with the main server over secure HTTPS connections to retrieve necessary data. Identity verification for managers is also performed via HTTPS to obtain validation results. The application accesses external APIs directly from the frontend without requiring secret keys, reducing unnecessary backend traffic and latency. This improves performance by eliminating the intermediate step and allows direct access to country, region, and city data via the CountriesNow API[19], geolocation coordinates via Nominatim[20], interactive map visualization with Leaflet[21], and image management with Cloudinary[22] for hotel units, rooms, and user profile images.

The main server forms the central backbone of the platform, responsible for processing client requests, managing data, and coordinating interactions with external services. It is implemented using Node.js and Express.js[23], providing a modular, asynchronous, and scalable environment suitable for high-volume operations.

Data persistence is achieved through MongoDB Atlas[24], with Mongoose ODM enabling structured schema definitions and simplified database interactions. The application architecture follows the Model–Router–Controller (MRC) pattern, separating concerns between data models, request routing, and business logic.

Security mechanisms include JWT-based authentication, with short-lived access tokens and longer-lived refresh tokens stored in HTTP-only cookies. Middleware enforces role-based access control, ensuring that users can only access resources corresponding to their role. High-privilege accounts, such as administrators and hotel managers, benefit from multi-factor authentication, while Google OAuth 2.0 integration allows convenient and secure login.

The server integrates several external APIs, services and methods to enhance functionality:

- ListaFirme[25] for verifying the legal status and sector compliance of hotels;
- OpenAI API[26] for generating automated hotel and room descriptions and providing personalized recommendations;
- Stripe[27] and APILayer for secure online payments and real-time currency conversion;
- Gmail API for email notifications related to registration, bookings, and payments;
- Cron jobs to automate reservation status updates and reminders.

Additionally, the server executes specialized algorithms, such as Last Minute Deals, to optimize hotel occupancy and user engagement. A centralized logging system records all significant actions, supporting auditing, diagnostics, and operational security.

In summary, the main server ensures the platform's core functionality, security, and extensibility, integrating external services seamlessly while maintaining high performance and reliability.

The Identity Verification Server is a dedicated component responsible for validating the authenticity of individuals registering accommodation units on the platform. Developed using Python and the Flask framework[28], the server ensures secure and automated authorization of hotel managers.

The server receives from the client a photograph of the identity card, three guided selfies, and user-provided data (first and last name). OCR techniques extract the textual information from the ID, which is then compared to the user-inputted data. Facial recognition algorithms subsequently compare the ID image with the selfies to verify identity. Anti-spoofing measures detect fraudulent attempts using printed photos, digital images, or videos.

At the client level, a guided capture system ensures correct data collection, employing MediaPipe FaceMesh[29] to detect facial landmarks in real-time. Sensitive data are encrypted with a hybrid encryption scheme, combining AES-GCM for data encryption and RSA-OAEP for session key protection, guaranteeing confidentiality during transmission.

The server processes incoming requests by decrypting the received data and validating ID information using Tesseract OCR[30], configured for Romanian and English, focusing on the MRZ (Machine-Readable Zone) for accurate extraction of names. Facial features are analyzed using the face_recognition library[31] based on Dlib[32], which employs CNN-based models pre-trained for robust detection across varying lighting conditions, expressions, and head orientations. The final step is anti-spoofing verification, designed to detect fake faces such as printed photos, digital images, or videos. This is achieved using Minivision Technology's Silent-Face-Anti-Spoofing[33], which operates without active user interaction, unlike gesture-based or client-side methods. The system uses a two-branch model—one for face classification and one for Fourier spectrum analysis—to detect subtle differences between real and fake faces. Results

[13] https://redux.js.org/
[14] https://reactrouter.com/
[15] https://restfulapi.net/
[16] https://axios-http.com/docs/intro
[17] https://fetch.spec.whatwg.org/
[18] https://getbootstrap.com/
[19] https://countriesnow.space/
[20] https://nominatim.org/
[21] https://leafletjs.com/
[22] https://cloudinary.com/
[23] https://expressjs.com/
[24] https://www.mongodb.com/products/platform
[25] https://www.listafirme.ro/specificatii/api-info-v2.asp
[26] https://openai.com/
[27] https://stripe.com/en-ro
[28] https://flask.palletsprojects.com/en/stable/
[29] https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/face_mesh.md
[30] https://github.com/tesseract-ocr/tesseract
[31] https://github.com/ageitgey/face_recognition
[32] https://dlib.net/
[33] https://github.com/minivision-ai/Silent-Face-Anti-Spoofing/blob/master/README_EN.md

 Digital Object Identifier 10.32754/JMT.2026.1.12

from the captured selfies are integrated into the decision process, with each image required to pass both facial and anti-spoofing checks to validate the user's identity.

This multi-layered verification process (secure data transmission, OCR-based ID validation, facial recognition, and anti-spoofing) constitutes a robust and reliable biometric authentication system. It ensures the platform registers only genuine hotel managers, preventing fraudulent accounts while maintaining a smooth and user-friendly experience.

IV. RESULTS

The implementation of the application demonstrates a comprehensive, role-based system capable of supporting complex hotel management and reservation processes. The results are analyzed with respect to database architecture, client interface functionality, manager tools, and administrative control, highlighting the effectiveness and robustness of the solution.

The solution focuses on three user types: administrator, hotel manager, and client, each with clearly defined roles and permissions. This structure enables the development of core functionalities that ensure operational efficiency while providing appropriate control over resources and reservations. The application is composed of three main components: a React client, delivering an intuitive and responsive interface; a Node.js server, handling business logic and resource management, and a Flask-based identity verification server, enhancing security and trust within the platform.

The database architecture (Fig. 3), implemented in MongoDB, provides a flexible and scalable structure based on key collections: Users, Hotels, Rooms, and Bookings. The Users collection organizes all user categories, differentiating roles through the user_type field and ensuring secure authentication and authorization. Hotels and Rooms collections manage detailed hotel and room data, including facilities, accessibility features, and availability. Auxiliary collections, such as Reviews, Unavailabilities, and Contacts, support complementary functionalities. Unique identifiers enable logical relationships between collections, ensuring data integrity and consistency within the NoSQL model.

Application security is ensured through HTTPS, encryption of sensitive data, and multifactor authentication for privileged accounts.

The client application allows users to perform personalized hotel searches, apply filters, view detailed room information (Fig. 4), and execute secure bookings. Dynamic updates of availability and real-time search suggestions enhance user experience.

Payment processing is integrated through Stripe, supporting multiple methods. Confirmation notifications ensure transparency and reliability (Fig. 5).

For hotel managers, the application provides tools for account creation, identity verification through multi-step authentication and biometric validation, hotel registration, room management, and reservation oversight. Features such as automated AI-assisted descriptions and stepwise multi-form interfaces streamline data entry and facilitate operational efficiency (Fig. 6). Managers can modify or cancel reservations, with automated updates and notifications sent to clients.

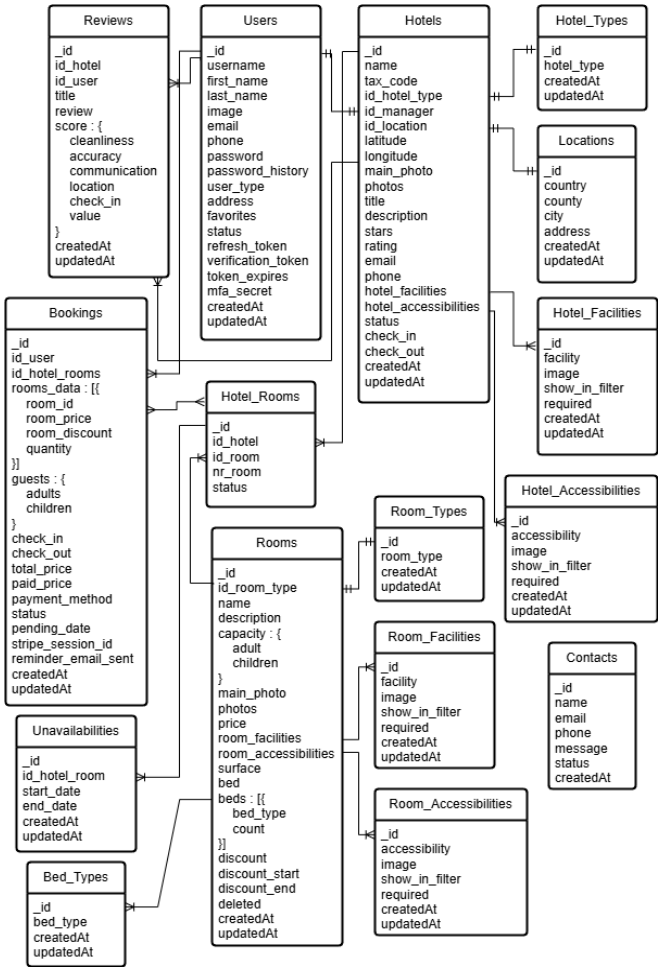

Figure 3. Logical relation diagram on database collections

The administrator interface centralizes platform oversight, enabling management of users, hotels, rooms, facilities, accessibility features, messages, notifications, and performance statistics. Administrative dashboards allow efficient control of all resources, monitoring of platform activity, and informed decision-making. The administrator's main menu, from the Romanian version of the page, depicts all of these (Fig. 7).

System testing was a crucial stage in the development process to ensure functional correctness and module integration. Unit testing focused on validating individual functionalities, including logic execution and error handling for various input scenarios. Integration testing assessed communication and collaboration between the client application, main server, and identity verification server, including external APIs. HTTP endpoints were validated using Insomnia[34], verifying CRUD operations, response formats, and authentication mechanisms. The identity verification module was tested under varying lighting conditions, demonstrating reliable performance. The user interface underwent responsiveness testing across different screen sizes to maintain visual coherence and accessibility.

Additionally, in order to evaluate system performances under peak loads, load and stress testing were conducted using Grafana K6[35]. The scalability and stability of the application were confirmed: it can handle significant user activity without performance degradation.

[34] https://insomnia.rest/
[35] https://github.com/grafana/k6

Overall, the implemented system successfully fulfills its objectives by ensuring secure and efficient hotel and room management, providing a seamless user experience, supporting robust administrative control, maintaining data integrity within a scalable NoSQL architecture, and demonstrating resilience under load conditions.

Integration with external services and APIs has expanded the platform's capabilities and ensured efficient and secure operations. The platform includes Leaflet for their interactive maps, CountriesNow for geolocation and different general information about countries and cities in simple JSON responses, Nominatim and OpenStreetMap data [21] to find locations on Earth by name and address and vice-versa, Google Authentication and Multi-Factor Authentication [22], OpenAI, and biometric technologies such as MediaPipe FaceMesh, Tesseract, and face_recognition. The adoption of open-source solutions reduced development costs while maintaining flexibility.

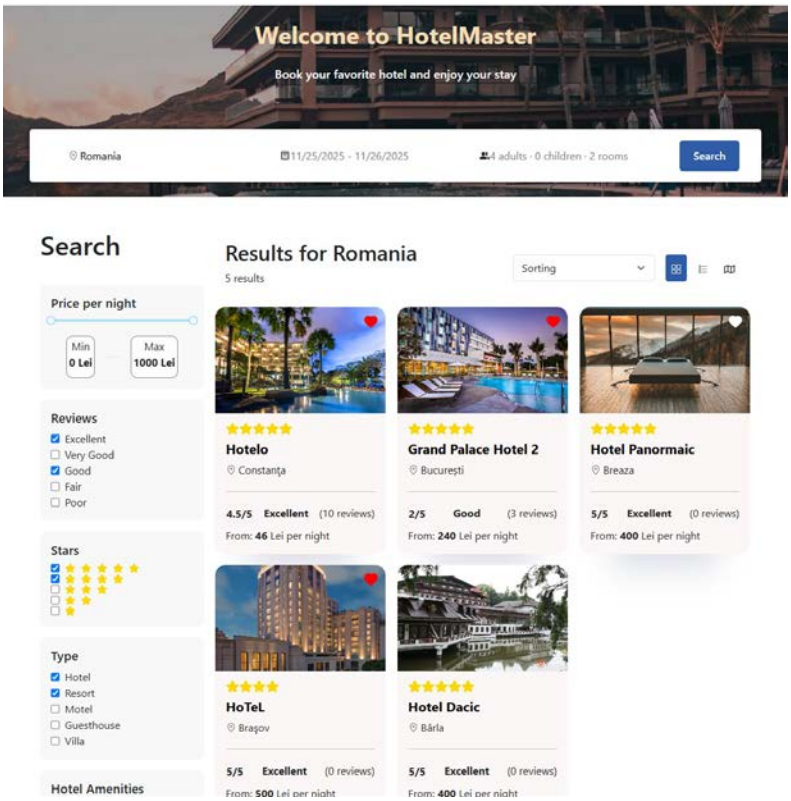

Figure 4. Hotel search page in client application

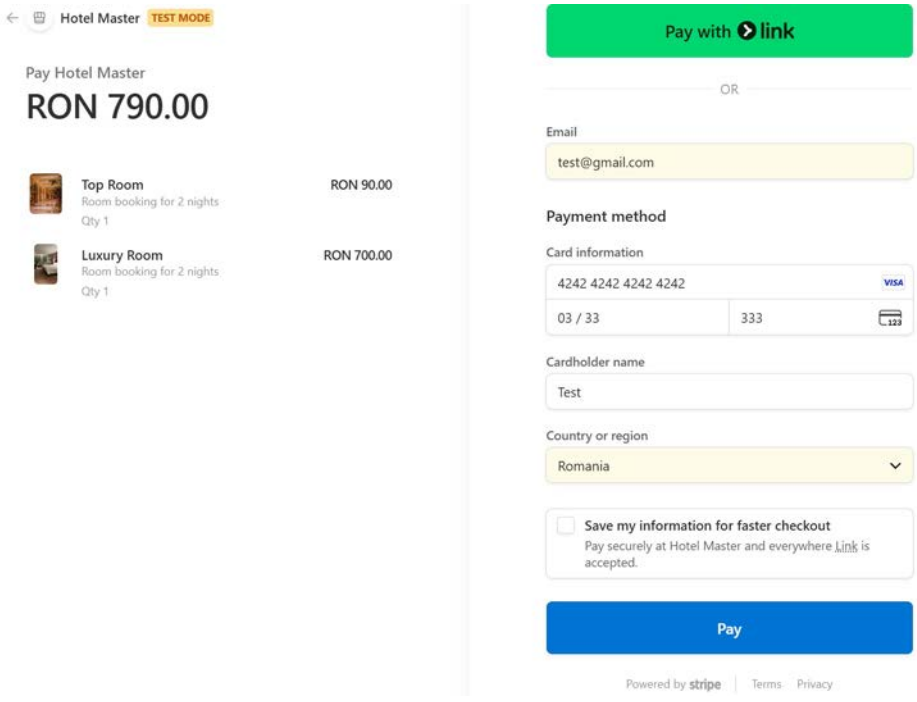

Figure 5. Payment page for credit card by Stripe

 Digital Object Identifier 10.32754/JMT.2026.1.12

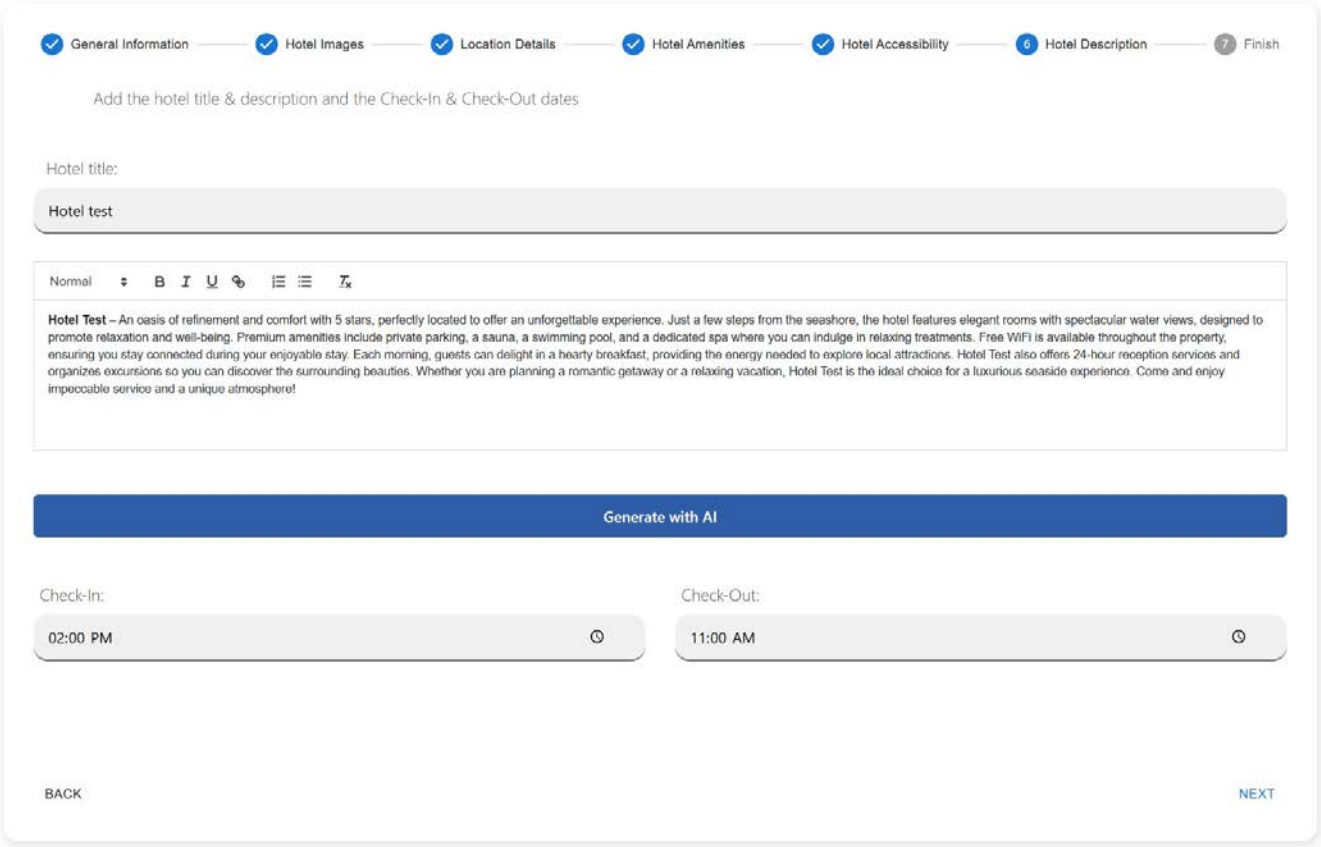

Figure 6. Manager's page to complete necessary data for the hotel

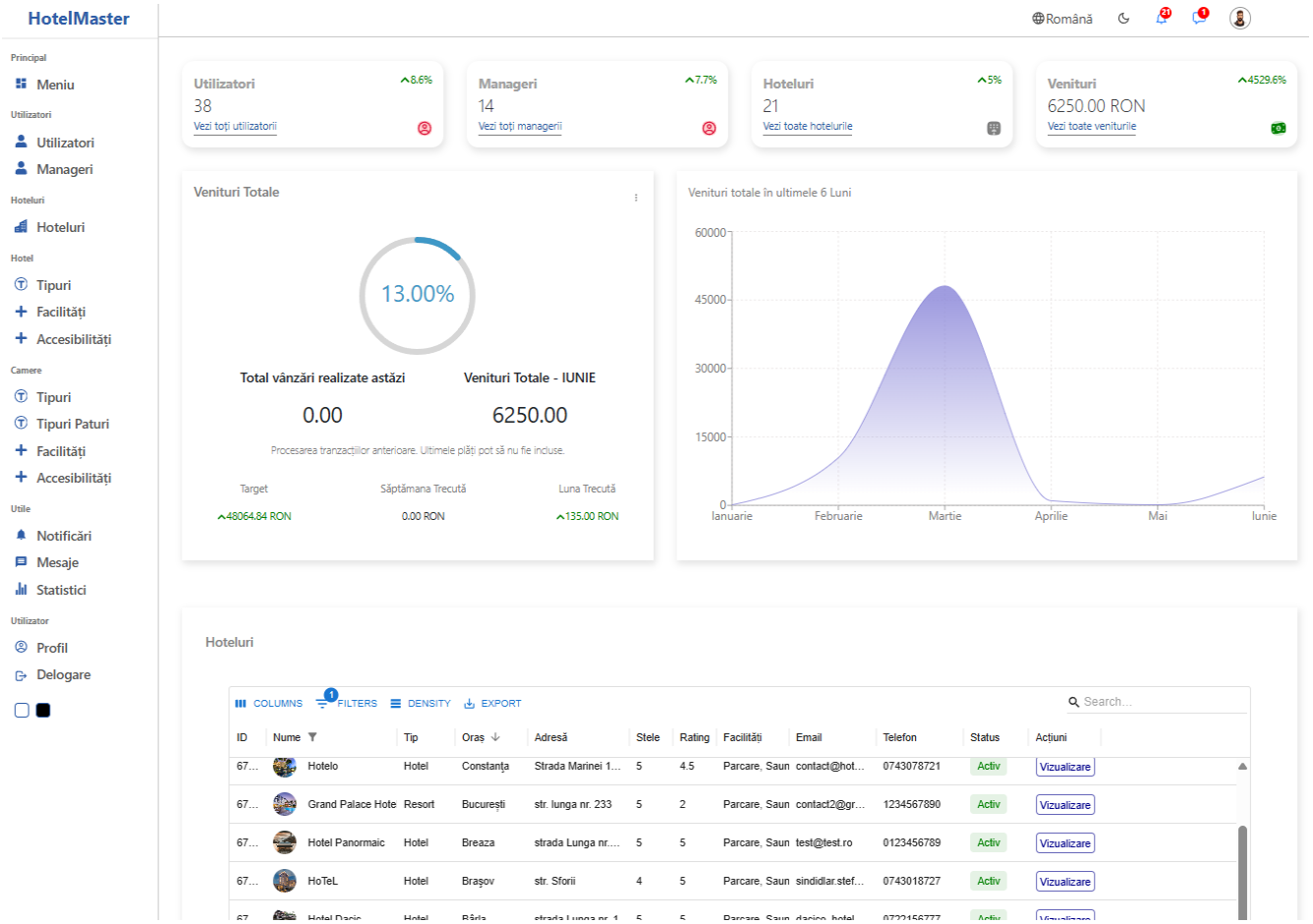

Figure 7. Romanian version of the administrator's main menu

V. CONCLUSION

In conclusion, the developed web platform demonstrates the feasibility of a modern, competitive solution for hotel resource management. By combining industry analysis with current technologies, robust security mechanisms, and scalability, the platform validates its potential to contribute to the digitalization and optimization of the hospitality sector. The application addresses the need for automated and efficient operations, reducing errors and enhancing user experience.

Future development directions include an AI-powered chatbot for client support, enabling direct messaging between clients and hotel staff, and integrating with centralized systems such as PMS and Channel Manager. These enhancements aim to optimize user experience and adapt to the dynamic demands of the hospitality market.

The current monolithic architecture facilitated rapid development and provided a solid foundation for future evolution, similar to successful industry platforms. Due to its complexity, we consider the hotel booking and resource management platform presented in this paper to be a good study case for the technologies used in complex web-based platforms.

REFERENCES

[1] L. Marinescu, S. Mîndroiu, and C. Ile, *Ghidul investitorului în industria hotelieră*. Sibiu, Romania: Agnos Publishing House, 2024.

[2] J. Hollander, "What is a hotel property management system (PMS): Complete guide," *Hotel Tech Report*, Feb. 2025. [Online]. Available: https://hoteltechreport.com/news/what-is-hotel-pms.

[3] D. Elphick, "Online travel agencies (OTAs): Full guide for hotels," *SiteMinder*, Sep. 2025. [Online]. Available: https://www.siteminder.com/r/best-otas-hotel-bookings/.

[4] S. Colcol, "OTA channel manager: The ultimate guide," *SiteMinder*, Mar. 2025. [Online]. Available: https://www.siteminder.com/r/ota-channel-manager/.

[5] M. Dawson, "Central reservation system (CRS): Guide for hotels," *SiteMinder*, Jun. 2025. [Online]. Available: https://www.siteminder.com/r/hotel-distribution/central-reservation-system/.

[6] D. Elphick, "Hotel booking engine: Complete guide for hotels," *SiteMinder*, Jun. 2025. [Online]. Available: https://www.siteminder.com/r/booking-engine-hotel/.

[7] J. Hollander, "Hotel CRM software: How to know what to look for in 2025," *Hotel Tech Report*, Jun. 2025. [Online]. Available: https://hoteltechreport.com/news/hotel-crm.

[8] "Revenue management systems for hotels: Products and features," *AltexSoft*, Jan. 2025. [Online]. Available: https://www.altexsoft.com/blog/revenue-management-system/.

[9] T. Şahin, "High-level system architecture of Booking.com," *Medium*, Jan. 2024. [Online]. Available: https://medium.com/@sahintalha1/high-level-system-architecture-of-booking-com-06c199003d94.

[10] Vintage Global, "Insight into the high level system architecture of Booking.com," *LinkedIn*, Jun. 2024. [Online]. Available: https://www.linkedin.com/pulse/insight-high-level-system-architecture-bookingcom-vintageglobal-hjoee/.

[11] "Securing your account," *Booking.com*. [Online]. Available: https://partner.booking.com/en-us/help/legal-security/security/securing-your-account.

[12] E. Veldhuizen, "Hospitality technology solutions for hotels," *Booking.com*. [Online]. Available: https://partner.booking.com/en-gb/hospitality-technology-solutions-hotels.

[13] J. Wallis, "Airbnb tech stack explained: The tech behind the Airbnb app," *Intuji*, Apr. 2023. [Online]. Available: https://intuji.com/airbnb-tech-stack-explained-airbnb-app/.

[14] J. Tai, "The human side of Airbnb's microservice architecture," *InfoQ*, Nov. 2021. [Online]. Available: https://www.infoq.com/presentations/airbnb-culture-soa/.

[15] ByteByteGo, "A brief history of Airbnb's architecture," *ByteByteGo Newsletter*, Feb. 2024. [Online]. Available: https://blog.bytebytego.com/p/a-brief-history-of-airbnbs-architecture.

[16] "Preferred software partners," *Airbnb*. [Online]. Available: https://www.airbnb.com/software-partners/preferred-partners.

[17] "Add and use two-factor authentication," *Airbnb*. [Online]. Available: https://www.airbnb.com/help/article/2842.

[18] "Facial recognition for identity verification," *Airbnb*. [Online]. Available: https://www.airbnb.com/help/article/3564.

[19] "Single-page application," *Wikipedia*, Oct. 2025. [Online]. Available: https://en.wikipedia.org/wiki/Single-page_application.

[20] "LocalStorage, sessionStorage," *JAVASCRIPT.INFO*, Oct. 2022, [Online]. Available: https://javascript.info/localstorage.

[21] "How good is OSM data?," *OpenStreetMap*. [Online]. Available: https://welcome.openstreetmap.org/working-with-osm-data/how-good-is-osm/.

[22] "Authentication methods at Google," *Google Cloud*. [Online]. Available: https://cloud.google.com/docs/authentication.