|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | | **Req#**  Requirements  -Functional Requirements | **Requirement** | **Description** | **Comments** | **Priority (Rete)** |  | | --- | --- | --- | --- | --- | --- | | **FR\_01** | User Authentication & Role Management | Users must log in with credentials and have access based on their role (Admin, Receptionist, Guest, Housekeeping, Restaurant Staff). | Must support multi-factor authentication (MFA) and password encryption. | High |  | | **FR\_02** | Room Booking & Availability System | Guests can search, view details, and book rooms online/offline. The system should show real-time availability. | Should allow guests to select room type, bed preferences, view, and amenities. | High |  | | **FR\_03** | Advanced Room Pricing Management | Admins should be able to set seasonal, promotional, or dynamic pricing for rooms. | Requires integration with analytics to adjust prices based on demand. | High |  | | **FR\_04** | Group Booking Feature | The system should allow bulk reservations for conferences, group travel, and corporate events. | Special discounts or customized invoices should be possible. | Medium |  | | **FR\_05** | Payment Processing & Invoicing | Guests should be able to pay using cash, credit/debit cards, PayPal, Stripe, or local bank integrations. | Secure payment gateway integration required, with invoice generation and automatic email receipts. | High |  | | **FR\_06** | Check-in & Check-out Process Automation | The system should allow QR code-based or self-service kiosk check-in/out to reduce manual workload. | Automated billing and payment processing should be integrated. | High |  | | **FR\_07** | Loyalty & Membership Program | The system should support guest loyalty programs, tracking points, discounts, and exclusive benefits. | Should allow automated discount application based on loyalty status. | Medium |  | | **FR\_08** | Guest Profile Management | Store guest information, preferences, past bookings, payment methods, and reviews. | Helps provide personalized services such as preferred room type and breakfast options. | High |  | | **FR\_09** | Housekeeping & Maintenance Management | Housekeeping should update room status (Cleaned, Needs Cleaning, Maintenance Issue). The system should track cleaning schedules and issue reports. | Integration with guest check-out for real-time updates. | Medium |  | | **FR\_10** | Restaurant, Bar & Room Service Ordering | Guests can order food, drinks, or room service via the system. | Bills should be automatically added to the guest’s invoice. | High |  | | **FR\_11** | Mini-Bar & Inventory Tracking | The system should track mini-bar consumption and automatically charge guests. | Should sync with the inventory system. | Medium |  | | **FR\_12** | Banquet Hall & Event Booking | Guests should be able to reserve conference rooms, wedding halls, or event spaces. | Should allow customized billing for events (catering, decorations, etc.). | Medium |  | | **FR\_13** | Parking Management System | The system should track guest parking reservations and vehicle details. | Should allow automated gate pass generation. | Medium |  | | **FR\_14** | Spa, Gym, and Additional Services Booking | Guests should be able to book spa treatments, gym access, or wellness sessions. | Can integrate with third-party services if needed. | Medium |  | | **FR\_15** | Lost & Found Management | The system should allow staff to record lost & found items and notify guests. | Tracking system for return requests. | Low |  | | **FR\_16** | Multi-Property Management | The system should allow admins to manage multiple hotel locations within one platform. | Useful for hotel chains. | High |  | | **FR\_17** | Multi-Language & Multi-Currency Support | The system should support different languages and allow guests to pay in various currencies. | Essential for international hotels. | High |  | | **FR\_18** | Reports & Business Intelligence Dashboard | The system should generate reports on occupancy, revenue, housekeeping, food sales, and customer satisfaction. | Should allow data exports (Excel, PDF) and graphical analytics. | High |  | | **FR\_19** | Automated Tax & Compliance Management | The system should calculate local taxes, VAT, service charges, and ensure regulatory compliance. | Should integrate with accounting software. | High |  | | |

### ****Non-Functional Requirements****

#### **Security**

* **Data Encryption (AES-256)**: All sensitive data, including personal guest information, payment details, and internal hotel data, should be encrypted using the AES-256 encryption standard. This ensures that any data breach or unauthorized access will not expose valuable personal or financial information.
  + **Compliance with Albania's Data Protection Law**: According to Albanian Law No. 9887/2008 and the new 2025 data protection law, encryption is mandatory for safeguarding personal data. This aligns with the GDPR and Albanian law to ensure data privacy.
* **Secure Login (OAuth2)**: User authentication should be managed using OAuth2, providing token-based secure access. This ensures that only authorized users (Admins, Receptionists, Guests, etc.) can access features based on their role. Multi-factor authentication (MFA) should be implemented for high-risk areas, such as admin access, payment processing, or modifying guest data.
  + **Compliance with GDPR and Albanian Law**: Strong authentication methods help ensure compliance with the GDPR's security principles, which emphasize the protection of personal data from unauthorized access.
* **Role-Based Access Control**: The system must incorporate role-based access control (RBAC), which ensures that each user has appropriate permissions to access only the parts of the system necessary for their role. For instance, receptionists should not have access to financial data, and guests should only see their booking information.
  + **Compliance with Albanian Law**: Properly enforced role-based access control supports the principles of data minimization and purpose limitation as per the data protection laws.
* **GDPR Compliance**: The system must ensure compliance with the General Data Protection Regulation (GDPR) and Albania's own data protection regulations (Law No. 9887/2008 and 2025 updates). This includes providing users with the ability to view, update, and delete their personal information, and implementing the right to be forgotten.

#### **Performance**

* **Concurrent User Support**: The system should be designed to support over 2000 concurrent users, including hotel staff, guests, and admin users. The system must be able to handle peak usage times, such as check-in/check-out periods, without significant performance degradation.
  + **Performance Requirements**: It should support high concurrency while ensuring a smooth and seamless user experience for all users without excessive waiting times or system crashes.
* **Response Time**: For critical operations (e.g., booking rooms, payment processing, or checking availability), the system should respond within 2 seconds. Fast response times are necessary to enhance user experience and avoid delays that could frustrate both guests and hotel staff.
* **Load Balancing**: The system should use load balancing to distribute incoming traffic evenly across multiple servers, ensuring optimal performance and preventing any server from becoming overwhelmed, especially during peak usage.

#### **Scalability**

* **Expansion to Multiple Locations**: The system should be designed to support the easy addition of new hotel locations as the business grows. It should be able to manage different properties under the same platform, each with its own room inventory, pricing strategies, and guest information.
  + **Multi-Database Sync**: The system should support multi-database synchronization to ensure that guest reservation data, availability, and room status are consistently updated across all locations.
* **Cloud-Ready**: The system should be cloud-compatible, enabling expansion to new locations or increased demand without requiring significant infrastructure changes. It should be able to scale resources up or down as needed without disrupting service.

#### **Usability**

* **Intuitive UI**: The user interface should be intuitive, easy to navigate, and designed to minimize training time. Hotel staff, regardless of their technical expertise, should be able to efficiently use the system for tasks such as booking, checking in guests, and managing room status.
  + **Compliance with Accessibility Standards**: The system should meet accessibility standards (e.g., WCAG) to ensure that users with disabilities can use the system effectively.
* **Mobile Support**: The system should be optimized for mobile use, allowing hotel staff to access key features (e.g., check-in/check-out, room availability) while on the go.
* **Training and Support**: Easy-to-access training materials should be available to hotel staff. The system should also include in-system tooltips, help sections, and video tutorials to assist users in getting started.

#### **Availability**

* **Uptime Guarantee**: The system should maintain a 99.9% uptime, ensuring that the system remains operational and accessible to hotel staff and guests almost all the time.
  + **Backup Systems**: There should be automated backup systems in place to provide redundancy in case of server failure.
* **Disaster Recovery**: The system should have a disaster recovery mechanism that ensures quick restoration of service in case of major outages, with minimal data loss.
* **Geo-Redundancy**: For systems serving multiple hotel locations, geo-redundancy should be implemented to ensure data availability in case one data center experiences issues.

#### **Reliability**

* **Auto-Backup**: The system should automatically back up data every 12 hours. These backups should include all guest information, booking details, room statuses, and payment history.
  + **Data Integrity**: The backup system should guarantee the integrity of data, ensuring that there is no corruption or loss during the backup process.
* **Recovery Options**: In case of system failure, reliable recovery options should be in place, such as point-in-time recovery, which will allow the system to be restored to the last known stable state.
* **Error Handling**: The system must include robust error-handling mechanisms to ensure that any unexpected issues, such as database failures or network issues, are properly logged and resolved without affecting the user experience.

#### **Maintainability**

* **Modular Architecture**: The system should use a modular architecture, where each component (e.g., booking, payment processing, guest management) operates independently. This allows easy updates and maintenance without disrupting the overall system.
* **Well-Documented Code**: The codebase should be well-documented, with clear comments and comprehensive documentation to assist future developers in making updates or addressing issues.
* **Automated Testing**: Automated testing should be in place to ensure that critical features of the system (e.g., booking engine, payment gateway) work as expected after each update or patch.
* **Regular Updates and Patches**: The system should be regularly updated to improve functionality, security, and compliance with changing regulations, particularly those related to data protection and e-commerce.

### ****Legal Compliance Considerations****

* **Data Protection**: Ensure the system is fully compliant with Albania's data protection laws, including the GDPR and the updated 2025 regulations. This includes proper handling of personal data, offering users the right to access, rectify, and delete their data, and implementing strong security measures like encryption and multi-factor authentication.
* **Consumer Protection**: The system should comply with Albanian consumer protection laws (e.g., Law No. 9902/2008), ensuring that all transactions are secure, transparent, and fair to guests.
* **Electronic Transactions**: The system should comply with the Albanian e-commerce regulations (Law No. 10128/2009) and electronic signature laws (No. 9880/2008), ensuring that electronic contracts, payments, and communications are legally binding.
* **HACCP Data Security**: All HACCP-related data (such as food safety logs, critical control point (CCP) monitoring, and temperature records) should be encrypted and securely stored. Only authorized staff should have access to this data, ensuring food safety protocols are strictly adhered to.

**Voucher and Receipt Management Functionalities**

1. **Voucher Issuance and Redemption**:
   * Guests should be able to use vouchers as discounts or credits toward their bookings, restaurant purchases, or spa services. Vouchers should be tracked in the system and linked to specific guest profiles or transactions.
   * Vouchers can be issued based on promotional campaigns, loyalty programs, or special events.
   * Vouchers should have specific rules regarding expiration, conditions for use, and applicable services.
2. **Voucher Validation**:
   * The system should validate vouchers during redemption by checking the expiration date, terms and conditions, and ensuring they haven't already been used.
   * The system should also notify users if a voucher is invalid (expired, redeemed, or non-existent).
3. **Receipt Scanning**:
   * Guests should be able to scan and submit receipts for payment verification or refunds through the system.
   * Staff should be able to use scanners or mobile devices to scan receipts, ensuring that the correct amount is charged to the guest.
   * Scanned receipts should be linked to the guest’s profile, allowing easy tracking for billing disputes, refund requests, or expense reports.
4. **Receipt Validation for Refunds and Claims**:
   * The system should allow staff to validate scanned receipts against guest transactions to process refunds or claims.
   * Scanned receipts should be processed and stored securely, and they should be able to trigger automated refund processes when necessary.

### ****User Requirements****

#### **A. Guest Requirements:**

1. **Guest Registration & Login:**
   * **Requirement**: Guests must be able to register an account on the system and log in with secure credentials.
   * **Details**: Guests will create an account with personal information such as name, contact details, and payment methods. They should also be able to use secure login features like email/password and multi-factor authentication (MFA).
2. **Room Search & Booking:**
   * **Requirement**: Guests should be able to search for available rooms based on their preferences (e.g., room type, bed preference, price range).
   * **Details**: The system should provide a real-time search of available rooms, including detailed descriptions, photos, prices, and amenities. Guests should be able to book a room directly through the website or app, making payments through various methods.
3. **Voucher Redemption:**
   * **Requirement**: Guests should be able to redeem vouchers for discounts or promotions at check-out.
   * **Details**: Vouchers can be applied to bookings, restaurant orders, or additional services. Guests will input a voucher code, and the system will validate its availability, validity, and eligibility for the applied discount.
4. **Check-In & Check-Out Process:**
   * **Requirement**: Guests should be able to perform check-in and check-out either at the reception desk or via self-service kiosks or mobile devices.
   * **Details**: The system should support a smooth check-in/check-out process, including verifying guest details, processing payments, and issuing room keys (physical or digital).
5. **Order Services (Food, Spa, etc.):**
   * **Requirement**: Guests should be able to order food, drinks, or book spa/gym services directly through the system.
   * **Details**: The system must allow guests to place orders from the restaurant, room service, or book additional services such as spa or gym time.
6. **Receipt and Invoice Generation:**
   * **Requirement**: Guests must receive receipts or invoices for payments, including any extra services such as restaurant charges or spa treatments.
   * **Details**: After check-out, the system should automatically generate a receipt or invoice for the guest’s payment history, which can be sent via email or available for download.

#### **B. Staff (Admin, Receptionist, Housekeeping, Restaurant Staff) Requirements:**

1. **User Role Management:**
   * **Requirement**: Staff must be assigned roles such as Admin, Receptionist, Housekeeping, Restaurant Staff, and have access to the relevant functionalities based on their role.
   * **Details**: Admins can manage all aspects of the system, while receptionists only manage bookings and check-ins. Housekeeping staff can mark room statuses, and restaurant staff can manage food orders.
2. **Room Management:**
   * **Requirement**: Staff should be able to view room availability, assign rooms, and update room statuses.
   * **Details**: Receptionists should have access to real-time room status updates (available, booked, cleaned, etc.). Housekeeping staff should update the room status to “cleaned,” and maintenance staff should indicate when rooms need repairs.
3. **Voucher Management:**
   * **Requirement**: Staff should be able to create, issue, and track vouchers.
   * **Details**: Admins can create promotional vouchers, while receptionists can apply them during check-out. The system should automatically validate the vouchers during redemption and prevent double usage.
4. **Guest Profile Management:**
   * **Requirement**: Staff should be able to access and update guest profiles, including personal information, booking history, and loyalty program status.
   * **Details**: Admins or receptionists can view and edit guest profiles for better service. This includes changing contact details or updating loyalty points.
5. **Reporting & Analytics:**
   * **Requirement**: Admins should have access to detailed reports on room bookings, financial data, service orders, voucher redemptions, and guest feedback.
   * **Details**: The system should provide graphical representations of occupancy rates, revenue, and performance metrics, including the effectiveness of promotional campaigns like voucher use.

### ****System Requirements****

**System Requirements** define the technical specifications and features of the system. These requirements describe the underlying **technology, architecture, and capabilities** needed to fulfill the user needs.

#### **A. Functional System Requirements:**

1. **User Authentication & Authorization:**
   * **Requirement**: The system must support user authentication and role-based access control.
   * **Details**: The system should authenticate users via username/password and provide role-based access. Admins should have full access to all data, while staff such as receptionists or housekeepers should only have access to their respective modules.
2. **Room Booking & Availability System:**
   * **Requirement**: The system should manage real-time room availability and allow guests to book rooms.
   * **Details**: The system should connect to a central database to track room occupancy and availability. It should prevent overbooking and allow for flexible bookings based on guest preferences.
3. **Voucher Management:**
   * **Requirement**: The system should allow the creation, tracking, and validation of vouchers.
   * **Details**: Admins can create promotional vouchers and set their conditions (validity period, discount percentage). When guests enter voucher codes, the system should validate them and apply the discounts automatically at check-out.
4. **Payment Gateway Integration:**
   * **Requirement**: The system must integrate with multiple payment gateways, such as credit/debit cards, PayPal, Stripe, etc.
   * **Details**: The system should allow guests to securely make payments online or at the hotel using various methods. It should handle transactions efficiently and issue receipts or invoices upon successful payment.
5. **Receipt Scanning & Management:**
   * **Requirement**: The system should allow the scanning and storing of receipts.
   * **Details**: Staff should be able to scan receipts for services and expenses related to a guest's stay. The scanned receipts should be stored in the guest’s profile and linked to their account for easy reference.
6. **Data Backup & Recovery:**
   * **Requirement**: The system must have automatic data backup and recovery mechanisms.
   * **Details**: The system should back up all critical data, including guest details, booking histories, payment records, and voucher information, to avoid loss in case of system failure.

#### **B. Non-Functional System Requirements:**

1. **Security:**
   * **Requirement**: The system must ensure that all sensitive data is encrypted and protected from unauthorized access.
   * **Details**: The system should use secure encryption protocols (e.g., AES-256) for data storage and communication. This is important for guest personal information and payment details.
2. **Performance:**
   * **Requirement**: The system should be capable of handling over 2000 concurrent users with fast response times.
   * **Details**: The system should ensure that operations such as booking, payment processing, and voucher redemption happen without delay, especially during peak usage.
3. **Scalability:**
   * **Requirement**: The system should scale to accommodate multiple hotel locations.
   * **Details**: The system should support multi-location management, allowing admins to manage different hotels under one account, with data synchronized across all properties.
4. **Availability:**
   * **Requirement**: The system should be available 99.9% of the time.
   * **Details**: This high availability ensures that the hotel’s operations, especially booking and check-in/check-out processes, are not interrupted.
5. **Usability:**
   * **Requirement**: The system should be user-friendly and accessible to both technical and non-technical staff.
   * **Details**: Staff should be able to quickly learn and navigate the system without intensive training, ensuring smooth operations and a good guest experience.