**Functional Requirements:**

**Functional Requirement 1: Booking Application User Interface**

**Functional Requirement 1.1: Direct Booking Interface**

**Test Case 1.1.A: Successful Room Booking**

Verify a guest can successfully book a room through the website or mobile app without calling the hotel:

**Testing Process:**

A. Navigate to the booking section on the hotel's website or app.

B. Search for available rooms by entering desired dates.

C. Select a room and proceed to booking.

D. Enter personal and payment information, then confirm the booking.

E. Review the booking confirmation received via email or app notification.

**Verification:** The guest receives a booking confirmation via email or app notification which accurately reflects the dates, room type, rate, and any special requests made by the guest.

**Test Case 1.1.B: Mandatory Fields Validation**

Ensure the booking process cannot be completed without filling in all mandatory fields:

**Testing Process:**

A. Attempt to book a room without entering all required personal and payment information.

B. Try to confirm the booking.

**Verification:** The system prevents the booking from being confirmed and prompts the user to fill in all mandatory fields.

**Functional Requirement 1.2: Room Comparison and Availability**

**Test Case 1.2.A: Compare Different Room Types**

Verify guests can compare different room types based on various criteria:

**Testing Process:**

A. Enter search criteria for dates and view available rooms.

B. Use the comparison feature to select different room types.

C. Review the comparative details presented by the system.

**Verification:** The system allows for easy comparison of rooms based on type, availability, amenities, and rates.

**Test Case 1.2.B: Detailed Room Descriptions and Images**

Ensure that detailed descriptions and images for each room type are viewable:

**Testing Process:**

A. Select a room type from the list of available options.

C. Access the detailed view to see descriptions and images of the room.

**Verification:** Each room type has detailed descriptions and multiple images that accurately represent the room.

**Functional Requirement 1.3: Centralized Reservation Management for Staff**

**Test Case 1.3.A: Accessing and Reviewing Reservations**

Confirm staff can access and review all reservations in real time:

**Testing Process:**

A. Staff member logs into the centralized dashboard.

B. Navigate to the reservations management section.

C. Reviews a list of all current reservations.

**Verification:** Staff can access and review detailed information for all reservations in real time.

**Test Case 1.3.B: Modifying a Reservation**

Test the ability of staff to make changes to reservations:

**Testing Process:**

A. Select an existing reservation from the dashboard.

B. Modify details of the reservation (e.g., change dates, room type).

C. Save the changes.

**Verification:** Changes are saved and updated across all booking channels in real time.

**Test Case 1.3.C: Managing Overbookings**

Evaluate the system’s ability to handle and prevent overbookings:

**Testing Process:**

A. Create a booking that exceeds the available inventory.

B. Observe how the system responds to the attempt.

**Verification:** The system prevents the booking from being confirmed and alerts the staff to the overbooking situation.

**Functional Requirement 1.4: Real-Time Inventory Update**

**Test Case 1.4.A: Inventory Update on New Booking**

Verify that room inventory updates in real-time when a new booking is made:

**Testing Process:**

A. Make a booking through the website or app.

B. Immediately check room availability for the same dates.

**Verification:** The inventory reflects the new booking by showing reduced availability.

**Test Case 1.4.B: Inventory Update on Cancellation**

Ensure room inventory is immediately updated upon cancellation:

**Testing Process:**

A. Cancel an existing booking.

B. Check the room availability for the previously booked dates.

**Verification:** The inventory is updated to reflect the cancellation, showing increased availability.

**Functional Requirement 2: Check-In/Out and Room Assignment Automation**

**Functional Requirement 2.1: Mobile Check-In and Check-Out**

**Test Case 2.1.A: Mobile Check-In Process**

Verify guests can complete the check-in process via mobile devices:

**Testing Process:**

A. Open the hotel's app and navigate to the check-in section.

B. Enter reservation details and confirm check-in.

C. Receive a digital room key on the mobile device, if applicable.

**Verification:** The guest successfully checks in through the app and receives a digital room key without needing to visit the front desk.

**Test Case 2.1.B: Mobile Check-Out Process**

Ensure guests can check out using their mobile devices:

**Testing Process:**

A. Use the hotel's app to initiate the check-out process.

B. Review and settle the bill electronically through the app.

C. Confirm check-out completion.

**Verification:** The guest completes the check-out process via the app, settles the bill electronically, and receives a confirmation of check-out.

**Test Case 2.1.C: Automatic Room Status Update for Housekeeping**

Ensure the system automatically updates the room status for housekeeping upon express check-out:

**Testing Process:**

A. Complete the express check-out process via the app.

B. Verify the room status is immediately updated in the housekeeping section of the system.

**Verification:** The room status is updated to "needs cleaning" immediately after the guest completes the express check-out, signaling housekeeping for service.

**Functional Requirement 2.2: Room Changes and Upgrades via Mobile**

**Test Case 2.2.A: Requesting Room Upgrades through the App**

Verify guests can request room upgrades through the hotel’s mobile app:

**Testing Process:**

A. Log into the app and navigate to the reservation details.

B. Select the option to view and request available upgrades.

C. Choose an upgrade and confirm the selection, noting any additional charges.

D. After requesting a room upgrade, check for a confirmation notification through the app or email.

**Verification:** The guest successfully requests a room upgrade through the app, with the new charges (if any) indicated, and receives confirmation of the room upgrade.

**Test Case 2.2.B: Accessibility of Upgraded Room**

Confirm that guests can access the upgraded room using the digital key provided:

**Testing Process:**

A. Use the digital room key provided post-upgrade to access the new room.

**Verification:** The digital room key grants access to the upgraded room as confirmed.

**Functional Requirement 2.3: Automated Room Assignment Based on Preferences**

**Test Case 2.3.A: Preference-Based Room Assignment**

Verify the system automatically assigns rooms based on guest preferences and loyalty status:

**Testing Process:**

A. Check-in via the mobile app, ensuring to list specific room preferences.

B. Observe the room assignment process and the final room allocated.

**Verification:** The room assigned matches the guest's preferences and takes into account their loyalty status, where applicable.

**Test Case 2.3.B: Modification of Room Assignment Pre-Arrival**

Test the system’s flexibility in modifying room assignments before guest arrival based on updated preferences or availability:

**Testing Process:**

A. After an initial room assignment, update preferences via the app.

B. Monitor any changes to the room assignment before arrival.

**Verification:** The system updates the room assignment according to the new preferences before the guest's arrival.

**Functional 3: Guest Profile Management**

**Functional Requirement 3.1: Access and Update Guest Profiles**

**Test Case 3.1.A: Updating Guest Profile**

Verify guests can update guest profiles with new preferences:

**Testing Process:**

A. Log into the system and access the guest profile.

B. Adds a new preference or special request to the guest profile.

**Verification:** The update is successfully saved, and the guest can view the added preferences or requests in the guest profile.

**Test Case 3.1.B: Viewing Guest Profile**

Ensure guests can view the history and preferences of a guest:

**Testing Process:**

A. Guest accesses the guest profile section.

B. Searches for the profile and reviews their stay history and preferences.

**Verification:** The guest can view detailed information about past stays, preferences, and special requests of the guest.

**Test Case 3.1.C: Profile Access Security**

Test the security and accessibility of guest profiles:

**Testing Process:**

A. Attempt to access the guest profile system with different accounts, including those with restricted access.

**Verification:** Only authorized guests can view and edit guest profiles, ensuring privacy and data security.

**Functional Requirement 3.2: Automatic Profile Updates**

**Test Case 3.2.A: Logging Dining Preferences**

Verify the system automatically updates guest profiles with dining preferences after a dining service request:

**Testing Process:**

A. A guest makes a dining service request specifying a dietary preference.

B. The system enters the request into the system.

**Verification:** The guest's profile is automatically updated with the new dining preference.

**Test Case 3.2.B: Service Request Logging**

Confirm every service request made by a guest is logged in their profile:

**Testing Process:**

A. Guest requests an additional service (e.g., spa service).

B. System logs the request in the system.

**Verification:** The guest's profile reflects the new service request in real time.

**Functional Requirement 3.3: Analyze Guest Feedback for Personalization**

**Test Case 3.3.A: Feedback Trend Analysis**

Test the system's capability to aggregate and display guest feedback trends:

**Testing Process:**

A. Manager accesses the guest feedback section in the system.

B. Reviews feedback trends and preferences aggregated by the system.

**Verification:** The system displays clear trends in guest feedback, which can be used for making service improvements and personalizing guest experiences.

**Test Case 3.3.B: Creation of Personalized Offers Based on Feedback**

Verify the system enables the manager to create personalized offers based on guest feedback:

**Testing Process:**

A. Identify a common preference or feedback trend.

B. Create a personalized offer addressing this trend.

**Verification:** The system allows for the creation and dissemination of offers tailored to identified preferences or feedback trends.

**Test Case 3.3.C: Feedback-Based Service Improvement**

Ensure guest feedback is used for continuous service improvement:

**Testing Process:**

A. Manager identifies a service area with negative feedback trends.

B. Implements changes based on this feedback.

**Verification:** Follow-up feedback indicates improvement in the targeted service area, demonstrating effective use of guest feedback for service enhancement.

**Functional 4: Hotel Management Automation**

**Functional Requirement 4.1: Dynamic Pricing Strategies**

**Test Case 4.1.A: Implementing Dynamic Pricing Rules**

Verify the revenue manager can set and implement dynamic pricing rules based on various factors:

**Testing Process:**

A. The revenue manager logs into the system and navigates to the pricing management section.

B. Set up new pricing rules based on occupancy, seasonality, and upcoming events.

C. Review the rates applied by the system under these conditions.

**Verification:** The system updates room rates across all platforms according to the new rules set by the revenue manager.

**Functional Requirement 4.2: Real-Time Pricing Adjustments**

**Test Case 4.2.A: Manual Rate Adjustment**

Verify manager can manually adjust rates based on real-time data:

**Testing Process:**

A. Identify a need for rate adjustment (e.g., competitor lowers prices).

B. manager manually adjusts the hotel's rates through the system.

**Verification:** The manual adjustments are reflected immediately across all booking channels.

**Test Case 4.2.B: Response to Market Changes**

Evaluate the system's capability to adjust pricing in response to market changes:

**Testing Process:**

A. Simulate a rapid change in market conditions (e.g., major event announcement).

B. Observe the system's response in adjusting rates.

**Verification:** The system promptly adjusts rates in response to new market data, maintaining competitive pricing.

**Functional Requirement 4.3: Booking Trend Analysis**

**Test Case 4.3.A: Identifying High-Demand Periods**

Ensure the system can analyze booking trends to identify high-demand periods:

**Testing Process:**

A. manager accesses the trend analysis tool in the system.

B. Reviews forecasts and reports generated by the system on booking trends.

**Verification:** The system accurately identifies upcoming high-demand periods, allowing for strategic pricing adjustments.

**Test Case 4.3.B: Monitoring Booking Trends Post-Adjustment**

Verify the system tracks the effectiveness of pricing adjustments on booking trends:

**Testing Process:**

A. After adjusting prices for a high-demand period, monitor booking rates and occupancy.

**Verification:** The system provides feedback on the effectiveness of the pricing strategy, showing an increase in bookings or occupancy as intended.

**Functional Requirement 4.4: Access to Dynamic Pricing Offers**

**Test Case 4.4.A: Displaying Dynamic Rates to Guests**

Verify guests can view and access dynamic pricing offers through the booking platform:

**Testing Process:**

A. A guest searches for rooms on the hotel's booking platform.

B. Observe the rates displayed, which should reflect dynamic pricing.

**Verification:** The most competitive rates, adjusted according to dynamic pricing algorithms, are visible to guests, allowing them to choose the best time and rate for their stay.

**Test Case 4.4.B: Booking Platform Rate Consistency**

Confirm that dynamic pricing rates are consistent across all user access points:

**Testing Process:**

A. Compare rates displayed on the hotel’s booking platform with those on mobile apps and partner websites.

**Verification:** Rates are consistent across all platforms, ensuring guests always receive the best offer available no matter how they access the booking system.

**Functional Requirement 4.5: Notification of Special Rates**

**Test Case 4.5.A: Special Rate Notifications Based on Preferences**

Verify the system notifies guests of special rates for their periods of interest based on stored preferences:

**Testing Process:**

A. The system identifies guests with specific stay preferences and historical booking patterns.

B. Sends personalized notifications about special rates matching their preferences.

**Verification:** Guests receive notifications (via email, SMS, or app) about special rates that align with their preferences, encouraging repeat bookings.

**Functional 5: Housekeeping and Maintenance Scheduling**

**Functional Requirement 5.1: Efficient Housekeeping Task Scheduling**

**Test Case 5.1.A: Assignment and Tracking of Housekeeping Tasks**

Ensure the housekeeping manager can efficiently assign and track housekeeping tasks:

**Testing Process:**

A. The housekeeping manager logs into the system and assigns cleaning tasks to staff, specifying rooms and deadlines.

B. The manager monitors the progress of these tasks in real time through the system.

**Verification:** The system accurately tracks the progress and completion of assigned tasks, ensuring rooms are prepared to the hotel's standards.

**Test Case 5.1.B: Adjusting Housekeeping Schedules**

Test the system's flexibility in adjusting housekeeping schedules based on changing guest needs or emergencies:

**Testing Process:**

A. An unexpected need for a room change arises, requiring immediate cleaning of a different room.

B. The housekeeping manager uses the system to reassign tasks to accommodate the new priority.

**Verification:** The system allows for quick adjustment of tasks, and staff are notified of the changes in real time, ensuring guest satisfaction.

**Functional Requirement 5.2: Real-Time Notifications for Room Status**

**Test Case 5.2.A: Notification upon Guest Check-Out**

Confirm that housekeeping staff receive immediate notifications when guests check out:

**Testing Process:**

A. A guest completes the check-out process.

B. Housekeeping staff immediately receive a notification that the room is now ready to be cleaned.

**Verification:** Notifications are promptly sent out, allowing housekeeping to prepare rooms for new arrivals efficiently.

**Test Case 5.2.B: Maintenance Request Notifications**

Ensure maintenance staff receive real-time notifications for Maintenance Request :

**Testing Process:**

A. A guest or staff member reports a maintenance issue through the system.

B. Maintenance staff receive an instant notification detailing the issue and location.

**Verification:** Maintenance requests are quickly communicated, allowing for timely repairs and improved room conditions.

**Functional Requirement 5.3: Scheduling Preventive Maintenance Checks**

**Test Case 5.3.A: Setting Up Preventative Maintenance Schedules**

Verify that maintenance staff can schedule and track preventative maintenance tasks:

**Testing Process:**

A. Maintenance staff schedules regular checks on hotel facilities and rooms within the system.

B. Staff follow through with checks and log completed work.

**Verification:** The system supports scheduling and documentation of preventative maintenance, ensuring the longevity and quality of the hotel's facilities.

**Test Case 5.3.B: Reminder Notifications for Upcoming Maintenance**

Test the system's ability to send reminders for scheduled preventative maintenance tasks:

**Testing Process:**

A. Maintenance staff set up reminders for upcoming maintenance activities.

B. As the scheduled dates approach, staff receive notification reminders.

**Verification:** Staff receive timely reminders, aiding in the proactive upkeep of the hotel's infrastructure and room quality.

**Functional Requirement 5.4: Guest-Requested Housekeeping Services**

**Test Case 5.4.A: Guest Requests for Housekeeping Services**

Ensure guests can request housekeeping services at their convenience through the app or website:

**Testing Process:**

A. A guest uses the hotel’s app to request housekeeping services for a specific time.

B. The housekeeping schedule is updated to reflect this request.

**Verification:** The system successfully schedules housekeeping according to the guest's request, enhancing their stay experience.

**Test Case 5.4.B: System Updates for Housekeeping Schedules**

Confirm that housekeeping schedules are automatically updated in response to guest requests:

**Testing Process:**

A. Housekeeping staff access their daily schedule after a guest makes a service request.

**Verification:** The schedule includes the new request at the specified time, showing the system's effective management of housekeeping tasks.

**Functional 6: Customer Relationship Management (CRM) Integration**

**Functional Requirement 6.1: Targeted Marketing Campaigns and Loyalty Programs**

**Test Case 6.1.A: Segmentation for Marketing Campaigns**

Validate the system's ability to utilize CRM data for creating segmented marketing campaigns:

**Testing Process:**

A. The marketing manager accesses CRM data to segment guests based on their stay history and preferences.

B. Creates a targeted marketing campaign for one of the segments.

**Verification:** The campaign is successfully sent to the targeted segment, demonstrating the system's effective use of CRM data for personalized marketing.

**Test Case 6.1.B: Loyalty Program Enrollment Offers**

Test the system's capability to send loyalty program invitations to guests based on CRM data:

**Testing Process:**

A. Identify guests who have stayed multiple times but are not enrolled in the loyalty program.

B. Send personalized loyalty program invitations via email or app notifications.

**Verification:** Identified guests receive the loyalty program invitation, indicating the system's ability to engage repeat guests further.

**Functional Requirement 6.2: CRM Data Analysis for Service Improvement**

**Test Case 6.2.A: Identifying Trends from Guest Data**

Ensure the system can analyze guest data to identify trends for service improvement:

**Testing Process:**

A. Hotel staff uses the system to generate reports on guest preferences and satisfaction scores.

B. Review the reports to identify areas for service improvement.

**Verification:** The system provides actionable insights, helping staff to make informed decisions on enhancing hotel services.

**Test Case 6.2.B: Implementing Improvements Based on CRM Insights**

Test the application of CRM insights for service improvement:

**Testing Process:**

A. Based on CRM data analysis, implement a new service or amend an existing one.

B. Monitor guest feedback and satisfaction scores post-implementation.

**Verification:** Improved guest satisfaction scores indicate the effective use of CRM insights for service improvements.

**Functional Requirement 6.3: Recognition of Returning Guests**

**Test Case 6.3.A: Automated Loyalty Rewards for Returning Guests**

Verify the system automatically recognizes returning guests and applies loyalty rewards:

**Testing Process:**

A. A returning guest makes a booking.

B. The system automatically identifies the guest as a loyalty member and applies available rewards or benefits.

**Verification:** The guest's booking reflects the applied loyalty rewards, confirming the system's capability to recognize and reward returning guests.

**Test Case 6.3.B: Personalized Welcome Back Messages**

Test the system's ability to send personalized welcome messages to returning guests:

**Testing Process:**

A. Upon check-in, a returning guest is identified by the system.

B. A personalized welcome back message is sent to the guest's mobile device or email.

**Verification:** The returning guest receives a tailored welcome message, enhancing their sense of value and appreciation.

**Functional 7: Feedback and Review Management**

**Functional Requirement 7.1: Systematic Collection and Analysis of Guest Feedback**

**Test Case 7.1.A: Collecting Feedback**

Ensure the system can systematically collect guest feedback from various sources (direct surveys, online platforms):

**Testing Process:**

A. Deploy a guest satisfaction survey post-stay via email and integrate feedback collection mechanisms on the hotel's app and website.

B. Aggregate collected feedback in the system.

**Verification:** Feedback from all channels is consolidated into the system, enabling comprehensive analysis.

**Test Case 7.1.B: Implementing Feedback-driven Improvements**

Verify the implementation of improvements based on guest feedback trends:

**Testing Process:**

A. Based on identified trends, implement specific service or operational improvements.

B. Monitor subsequent guest feedback for changes in satisfaction levels.

**Verification:** Improvement in guest satisfaction in previously identified areas confirms the effective use of feedback for service enhancement.

**Functional Requirement 7.2: Responsive Online Review Management**

**Test Case 7.2.A: Monitoring and Responding to Online Reviews**

Ensure the system enables staff to monitor and respond to online reviews promptly:

**Testing Process:**

A. Aggregate online reviews from various platforms into the system.

B. Use the system to respond to reviews, address concerns, and thank guests.

**Verification:** The system facilitates timely responses to reviews, helping to manage the hotel's online reputation effectively.

**Test Case 7.2.B: Tracking Response Impact**

Test the system's ability to track the impact of staff responses on online reputation:

**Testing Process:**

A. Analyze the correlation between response activities and changes in review scores or sentiment.

**Verification:** Positive trends in review scores or guest sentiment post-response indicate successful online reputation management.

**Functional Requirement 7.3: Streamlining Feedback Submission**

**Test Case 7.3.A: Ease of Feedback Submission**

Verify that guests can easily provide feedback during or after their stay through preferred channels:

**Testing Process:**

A. Guests are prompted to submit feedback via email, app, or website after their stay.

B. Evaluate the ease and completion rate of feedback submissions.

**Verification:** High completion rates and positive guest feedback on the submission process indicate its ease and effectiveness.

**Test Case 7.3.B: Direct Feedback Encouragement**

Ensure the system encourages guests to share their experiences directly with the hotel:

**Testing Process:**

A. Implement system prompts or incentives for guests to share their feedback directly.

B. Measure the increase in direct feedback submissions post-implementation.

**Verification:** An increase in direct feedback submissions confirms the system's effectiveness in encouraging guest participation.

**Functional 8: Data Analytics for Customer Insights**

**Functional Requirement 8.1: In-depth Analysis of Guest Behavior and Preferences**

**Test Case 8.1.A: Comprehensive Guest Data Analysis**

Validate the system's capability to analyze guest behavior, preferences, and trends comprehensively:

**Testing Process:**

A. Aggregate data from various guest touchpoints into the analytics module of the system.

B. Run an analysis to identify patterns and trends in guest behavior and preferences.

**Verification:** The system generates detailed reports highlighting key insights into guest behaviors and preferences, enabling targeted service enhancements and personalized guest experiences.

**Test Case 8.1.B: Utilization of Analytics for Service Personalization**

Test how effectively the hotel staff can use analytics insights to personalize guest services:

**Testing Process:**

A. Based on analytics insights, identify specific guest preferences for personalized service.

B. Implement personalized services for guests identified by the system.

**Verification:** Guests acknowledge the personalized touches to their service, indicating the successful application of analytics for service enhancement.

**Test Case 8.1.C: Predictive Analytics for Guest Satisfaction**

Ensure the system can use predictive analytics to forecast guest needs and enhance satisfaction:

**Testing Process:**

A. Utilize the system's predictive analytics feature to forecast guest needs based on historical data.

B. Proactively address these needs during the guest's stay.

**Verification:** Increased guest satisfaction scores post-visit, suggesting the effective use of predictive analytics in improving the guest experience.

**Functional Requirement 8.2: Predictive Booking Trends for Optimal Staffing and Resource Allocation**

**Test Case 8.2.A: Forecasting Future Booking Trends**

Verify the system's accuracy in forecasting future booking trends to optimize staffing levels and resource allocation:

**Testing Process:**

A. Access the system's forecasting tool to predict future booking trends.

B. Adjust staffing levels and resource allocation based on these forecasts.

**Verification:** Observations of optimal staffing and resource utilization during predicted peak and off-peak periods indicate the system's forecasting accuracy.

**Test Case 8.2.B: Response to Predictive Occupancy Changes**

Test the system’s flexibility in adjusting operational plans based on updated booking trend forecasts:

**Testing Process:**

A. Monitor changes in booking trend forecasts.

B. Observe how quickly the hotel can adjust staffing and resources in response to these updates.

**Verification:** The ability to quickly adapt to updated forecasts, avoiding both understaffing during peaks and wastage of resources during lulls, confirms the system's effectiveness.

**Functional Requirement 8.3: Benchmarking Guest Satisfaction**

**Test Case 8.3.A: Comparing Guest Satisfaction Against Industry Benchmarks**

Ensure the system can benchmark the hotel's guest satisfaction metrics against industry standards:

**Testing Process:**

A. Collect guest satisfaction data via surveys and feedback tools within the system.

B. Use the system to compare these metrics against known industry benchmarks.

**Verification:** The system provides a clear comparison, highlighting areas of strength and opportunities for improvement relative to industry peers.

**Test Case 8.3.B: Implementing Improvements Based on Benchmark Comparisons**

Test the system's capability to aid in focusing service improvement efforts based on benchmarking results:

**Testing Process:**

A. Identify specific areas lagging behind industry benchmarks.

B. Implement targeted service improvements.

**Verification:** Post-implementation feedback and metrics show improvement in previously identified areas, indicating the effective use of benchmarking for service enhancement.

**Functional Requirement 8.4: Ethical Handling of Guest Data**

**Test Case 8.4.A: Compliance with Data Protection Regulations**

Verify that the system's handling of guest data complies with applicable data protection regulations:

**Testing Process:**

A. Review the system's data handling procedures for compliance with GDPR or other relevant regulations.

B. Conduct a compliance audit or review.

**Verification:** The system passes the compliance review without any violations, ensuring guest data is handled ethically and securely.

**Test Case 8.4.B: Guest Data Security Measures**

Test the security measures implemented by the system to protect guest data:

**Testing Process:**

A. Perform security testing on the system's data storage and processing capabilities.

B. Check for vulnerabilities and the effectiveness of encryption, access controls, and other security measures.

**Verification:** The system demonstrates robust security against potential breaches, ensuring the protection of guest data.

**Non-Functional Requirements:**

**Requirement 1: Intuitive User Interface**

**Test Case 1.1: Usability Testing with Diverse Participants**

*Testing Process:*

A. Conduct a usability testing session involving at least 50 participants, including both guests and hotel staff.

B. Participants perform a series of tasks using the system, and their experiences are measured using the System Usability Scale (SUS).

*Verification:* The average SUS score from participants is at least 85, indicating a high level of usability.

**Test Case 1.2: Task Completion Rate**

*Testing Process:*

A. Measure the completion rate for a set of predefined tasks by participants during usability testing.

*Verification:* At least 95% of tasks are completed successfully by participants, demonstrating the system’s intuitive design.

**Test Case 1.3: User Feedback Collection**

*Testing Process:*

A. Collect qualitative feedback from participants on the system's user interface and ease of use.

*Verification:* Feedback indicates a positive user experience, with suggestions for improvements used to refine the interface further.

**Requirement 2: System Performance under Load**

**Test Case 2.1: Simulated Load Testing**

*Testing Process:*

A. Simulate up to 10,000 simultaneous users interacting with the system to test response times.

*Verification:* 95% of transactions were completed within 2 seconds under peak load conditions.

**Test Case 2.2: Real-Time Performance Monitoring**

*Testing Process:*

A. Implement real-time performance monitoring during peak operational hours.

*Verification:* The system maintains expected performance levels, with response times consistently under 2 seconds.

**Test Case 2.3: Stress Testing**

*Testing Process:*

A. Incrementally increase the load on the system until it reaches and surpasses expected peak levels.

*Verification:* System performance degrades gracefully beyond the peak load of 10,000 users, without catastrophic failure.

**Requirement 3: Data Security and Compliance**

**Test Case 3.1: Encryption Verification**

*Testing Process:*

A. Verify that all personal and payment information is encrypted both in transit and at rest using TLS encryption.

*Verification:* Encryption standards are met, with no data exposed in plain text during data transmission or storage.

**Test Case 3.2: Compliance Assessment**

*Testing Process:*

A. Conduct an internal audit to assess compliance with PCI DSS and GDPR.

*Verification:* The system passes the compliance assessment with no critical findings, ensuring adherence to legal standards for data protection and payment processing.

**Test Case 3.3: Security Penetration Testing**

*Testing Process:*

A. Perform penetration testing to identify vulnerabilities in the system’s security posture.

*Verification:* No critical vulnerabilities are found, and any minor issues are resolved, ensuring robust protection against potential cyber threats.

**Requirement 4: System Scalability**

**Test Case 4.1: Simulated User Growth**

*Testing Process:*

A. Simulate a 50% increase in user load to test the system's scalability.

*Verification:* The system handles increased load without significant degradation in performance or usability.

**Test Case 4.2: Resource Utilization Analysis**

*Testing Process:*

A. Monitor system resources (CPU, memory, database) under increased load conditions.

*Verification:* Resource utilization remains within optimal levels, indicating the system's ability to scale effectively.

**Test Case 4.3: Dynamic Resource Allocation**

*Testing Process:*

A. Test the system's ability to automatically allocate additional resources in response to increased demand.

*Verification:* The system dynamically scales up resources, maintaining performance levels without manual intervention.

These test cases address critical non-functional aspects like usability, performance, security, and scalability, ensuring the hotel management system meets the highest standards for data security, accessibility, and availability, enhancing overall user satisfaction.

**Requirement 5: System Reliability and Uptime**

**Test Case 5.1: Continuous Uptime Monitoring**

*Testing Process:*

A. Implement continuous monitoring of the system to track uptime over 12 months, excluding scheduled maintenance windows.

*Verification:* The system reports 99.9% uptime, demonstrating high reliability and minimal disruptions to hotel operations and guest experiences.

**Test Case 5.2: Recovery Time Objective (RTO) Testing**

*Testing Process:*

A. Simulate various failure scenarios to test the system's recovery capabilities and measure the time taken to return to full operational status.

*Verification:* The system recovers from failures within predefined RTO parameters, ensuring minimal impact on users.

**Test Case 5.3: Failover and Redundancy Verification**

*Testing Process:*

A. Test the system's failover mechanisms by simulating failures in primary services to observe automatic switches to redundant systems.

*Verification:* Failover processes activate seamlessly, with no noticeable disruption to users, verifying the system's reliability measures.

**Requirement 6: Cross-Platform and Browser Compatibility**

**Test Case 6.1: Major Web Browser Compatibility Testing**

*Testing Process:*

A. Test the system on the latest two versions of major web browsers (Chrome, Firefox, Safari, Edge) to ensure all core functionalities work without issues.

*Verification:* The system is fully operable and displays consistently across all tested browsers, confirming wide accessibility and functional integrity.

**Test Case 6.2: Mobile Platform Compatibility**

*Testing Process:*

A. Conduct mobile compatibility testing on devices running the latest two versions of iOS and Android.

*Verification:* The system's user interface and functionalities are fully operable on both iOS and Android platforms, ensuring accessibility for users on mobile devices.

**Test Case 6.3: Compatibility Issue Resolution**

*Testing Process:*

A. Identify and document any compatibility issues during testing.

B. Resolve identified issues within a defined period, not exceeding 30 days.

*Verification:* All compatibility issues are resolved promptly, ensuring the system provides a seamless experience across all user devices and platforms.

These test cases comprehensively cover the non-functional requirements essential for delivering a reliable, secure, maintainable, and user-friendly hotel management system.