**Hotel Room Booking System Specification**

**I. Functional Requirements**

The Hotel Room Booking System supports customers in searching, viewing information, booking rooms, making payments, and managing their reservations.  
It also assists receptionists or administrators in managing rooms, bookings, users, and tracking room usage status within the hotel.

The system is designed for standalone hotels or small chains (2–3 branches), operating on a moderate scale but expandable when needed.

**1. View and Search Rooms**

* Users can view the list of rooms by type (Standard, Deluxe, Suite, Family, VIP, etc.).
* The list can be sorted by price, maximum occupancy, rating, or availability.
* Rooms can be searched by name, check-in/check-out date, number of guests, price range, or room type.
* Each room displays images, a brief description, amenities, and nightly rate.
* When selecting a room, users can view detailed information including large images, full description, amenities, availability, and reviews from other guests.

**2. Customer Account Management (Member)**

* Users can register customers account with details: name, gender, date of birth, email, phone number, password, and address.
* They can log in, change passwords, and update personal information (except email).

When a password is forgotten, the system automatically generates a new one and sends it via email.

**3. Payment**

* Supports two payment methods:
  + **Pay at Hotel** – customers pay upon arrival.
  + **Online Payment** – via payment gateway (e-wallets, bank cards, QR Pay).
* Once payment is successful, the booking status changes to “Confirmed.”
* In case of cancellation, refunds are processed according to the hotel’s refund policy (if applicable).

**4. Room Booking**

* After choosing a room and stay dates, customers fill in booking information (name, phone number, notes, etc.).
* The system checks room availability and calculates the total amount.
* Customers can confirm their booking and choose a payment method.
* Once booked successfully, the system records the reservation and sends an email or notification to confirm.

**5. Booking Management**

* Users can view the list of their past and current bookings.
* Each booking includes: booking ID, check-in/check-out dates, room type, total cost, and status (New, Confirmed, Checked-in, Completed, Cancelled).
* Users can cancel bookings if the status is “New” and before the check-in date.

**6. System Administration (Admin/Receptionist)**

* **Room Management:** add, edit, delete, and update room status (available, being cleaned, occupied).
* **User Management:** view, add, edit, or delete customer accounts.
* **Booking Management:** view, confirm, modify, or cancel bookings.
* **Reports & Statistics:** track revenue, occupancy rate, and number of bookings over time.

**II. Non-Functional Requirements**

**1. Usability**

* The interface should be simple and easy to understand for both guests and receptionists.
* Fully responsive on desktop, mobile, and tablet.
* Booking process should be short—maximum 3 steps (Select Room → Enter Info → Confirm).

**2. Reliability**

* The system should operate stably during business hours or 24/7 as needed.
* Reliability ≥ 98–99%. Data should be backed up daily or weekly.

**3. Performance**

* Response time for main operations < 3 seconds.
* The system handles up to 200 concurrent requests.
* Storage capacity: 30,000–50,000 accounts and 100,000 bookings.
* Scalable to support additional hotel branches in the future.

**4. Security**

* User data and passwords are encrypted.
* Secure login/logout to prevent unauthorized access.
* Basic access control: Customer and Admin (Receptionist).

**5. Design Constraints**

* Application runs on a web or web app platform.
* Recommended technologies: C++, Python, Java.

**6. Legal & Privacy Compliance**

* Complies with the Vietnam Cybersecurity Law and personal data protection regulations.
* Customer information (name, phone, email, stay history) is used only for hotel service purposes.
* No customer data is shared with third parties without consent.

**III. Data Flow Diagram:**

A diagram of a hotel room

AI-generated content may be incorrect.

A diagram of a customer

AI-generated content may be incorrect.

**IV. Use-case diagram:**

1. **Sơ đồ tổng quan:**

A diagram of a software company

AI-generated content may be incorrect.

* 1. **List of actors:**

|  |  |  |
| --- | --- | --- |
| **Orders** | **Actor** | **Functional Description / Requirement Specification** |
| 1 | User | A non-registered or not-logged-in person who can browse and search rooms, and view room details, but cannot make bookings. |
| 2 | Customer | A registered and logged-in user who can search and view room details, book rooms, make payments, cancel bookings, provide cancellation reasons, view booking history, and update personal profile. |
| 3 | Administrator | The system administrator with full privileges. Responsible for managing customers, managing rooms (add, update, delete), managing bookings, viewing customers’ booking histories, and viewing system reports. |
| 4 | Payment | An internal or third-party payment service responsible for processing transactions, updating payment status, and handling refunds for canceled bookings. |

**1.2 List of Use-case:**

|  |  |  |
| --- | --- | --- |
| **Orders** | **Use Case** | **Functional Description / Requirement Specification** |
| 1 | Register | Allows a new user to create an account to access the system. |
| 2 | Login | Allows a user to log in with valid credentials. The session remains active until the user logs out or the session times out. |
| 3 | Search Room | Enables users to search for available rooms based on filters such as date, room type, or price (Extends: View Room Detail). |
| 4 | View Room Detail | Displays detailed information about a selected room (type, price, facilities, description, etc.). |
| 5 | Book Room | Allows customers to reserve a room for selected dates. (Includes: Process Payment, Update Booking Status). |
| 6 | Process Payment | Handles the processing of customer transactions when booking a room (Includes: Update Payment Status). |
| 7 | Update Payment Status | Updates the status of a payment (successful, failed, refunded, etc.). |
| 8 | Update Booking Status | Updates the status of a booking (booked, confirmed, canceled, completed, etc.). |
| 9 | Cancel Booking | Allows customers to cancel an existing booking (Includes: Provide Reason, Update Booking Status), and (Extends: Refund Payment). |
| 10 | Provide Reason | Collects the reason from the customer when canceling a booking (Included from Cancel Booking). |
| 11 | Refund Payment | Processes a refund after a booking is canceled (Extended from Update Payment Status). |
| 12 | Update Profile | Allows customers to edit their personal information (name, phone number, email, password, etc.). |
| 13 | Manage Customer | Enables the Administrator to view, edit, or delete customer accounts. |
| 14 | Add Room | Allows the Administrator to add a new room into the system (Extended from Manage Room). |
| 15 | Update Room | Allows the Administrator to modify an existing room’s details (Extended from Manage Room). |
| 16 | Delete Room | Allows the Administrator to remove a room from the system (Extended from Manage Room). |
| 17 | Manage Booking | Enables the Administrator to manage customer bookings (Extends: View Booking Customers’ Histories). |
| 18 | Manage Room | Allows the Administrator to manage and maintain all room-related data (Extends: Add, Update, Delete Room). |
| 19 | View Booking History | Allows customers to view their personal booking history. |
| 20 | View Booking Customers’ Histories | Allows the Administrator to view all customers’ booking histories. |
| 21 | View Reports | Allows the Administrator to view or export system reports (revenue, booking statistics, payment summaries, etc.). |

**2. Actor: User**

**2.1. Use case: Register**

**2.1.1. Description:** User can create a new account to access the system.

**2.1.2. Preconditions:**

* The user has not registered before.

**2.1.3. Postconditions:**

* A new account is created successfully.

**2.1.4. Main Flow:**

* User fills in personal information.
* System validates the input.
* Account is created and saved.
* System confirms successful registration.

**2.2. Use case: Login**

**2.2.1. Description:** User can log in to the system using valid credentials. Once logged in, the session remains active, allowing the user to continue using the system without re-login until logging out or session timeout.

**2.2.2. Preconditions:**

* User already has a registered account.
* No active session currently exists.

**2.2.3. Postconditions:**

* User session is created and kept active.

**2.2.4. Main Flow:**

* User accesses the login page.
* If a session exists, the system automatically redirects to the homepage.
* If not, user enters email and password.
* System validates credentials and logs in the user.
* User is redirected to the homepage.

**2.3. Use case: Search Room**

**2.3.1. Description:** User can search for available rooms based on preferences such as date, type, or price.

**2.3.2. Extensions:**

* <<extend>> View Room Detail – when user selects a specific room.

**2.3.3. Main Flow:**

* User enters search criteria.
* System displays a list of available rooms.
* User may view more details about a room.

**2.4. Use case: View Room Detail**

**2.4.1. Description:** User can view detailed information of a selected room, including type, price, and facilities.

**2.4.2. Trigger:**

* Extended from Search Room.

**2.5. Use case: Update profile**

**2.5.1. Description:** User can edit and update their personal details such as name, phone number, or password.

**2.5.2. Preconditions:**

* User is logged in.

**2.5.3. Postconditions:**

* Updated profile information is saved successfully.

**3. Actor: Customer ( inherits from User)**

**3.1. Use case: Book Room**

**3.1.1. Description:** Customer can book an available room for specific dates and complete payment online.

**3.1.2. Preconditions:**

* Customer is logged in.
* Room is available.

**3.1.3. Includes:**

* Process Payment
* Update Booking Status

**3.1.4. Main Flow:**

* Customer selects a room and booking date.
* System checks room availability.
* Customer proceeds with payment.
* System confirms booking and updates status.

**3.2. Use case: Cancel Booking**

**3.2.1. Description:** Customer can cancel a previously booked room.

**3.2.2. Preconditions:**

* Customer has an active booking.

**3.2.3. Includes:**

* Provide Reason
* Refund Payment
* Update Booking Status

**3.2.4. Main Flow:**

* Customer selects booking to cancel.
* Customer provides reason.
* System processes refund.
* System updates booking status.
* Cancellation confirmation is displayed.

**3.3. Use case: Provide Reason**

**3.3.1. Description:** Customer can provide the reason for cancelling a booking.  
**3.3.2. Trigger:**

* Included from Cancel Booking.

**3.4. Use case: Refund Payment**

**3.4.1. Description:** System can refund the payment amount to the customer’s account.  
 Trigger: Extend from Cancel Booking.  
**3.4.2. Includes:**

* Update Payment Status

**3.5. Use case: View Booking History**

**3.5.1. Description:** Customer can see the booking records of customers.  
**3.5.2. Trigger:**

* Extended from Manage Booking.

**4. Actor: Payment**

**4.1. Use case: Process Payment**

**4.1.1. Description:** Payment actor can process transactions when customers book a room.

**4.1.2. Includes:**

* Update Payment Status

**4.1.3. Main Flow:**

* System sends a payment request.
* Payment actor processes transaction.
* System updates payment status.
* Customer receives confirmation.

**4.2. Use case: Update Payment Status**

**4.2.1. Description:** Payment actor can update the payment record (success, failure, refund).

**4.2.2. Trigger:**

* Included from Process Payment or Refund Payment.

**5. Actor: Administrator**

**5.1. Use case: Manage Customer**

**5.1.1. Description:** Administrator can view, edit, and remove customer accounts.

**5.1.2. Preconditions:**

* Admin is logged in.

**5.1.3. Postconditions:**

* Customer information is updated or deleted as requested.

**5.2. Use case: Manage Room**

**5.2.1. Description:** Administrator can manage room information in the system.

**5.2.2. Extensions:**

* Add Room
* Update Room
* Delete Room

**5.3. Use case: Add Room**

**5.3.1. Description:** Administrator can add new room details to the system.  
**5.3.2. Trigger:**

* Extended from Manage Room.

**5.4. Use case: Update Room**

**5.4.1. Description:** Administrator can modify details of an existing room.  
**5.4.2. Trigger:**

* Extended from Manage Room.

**5.5. Use case: Delete Room**

**5.5.1. Description:** Administrator can remove room data from the system.  
**5.5.2. Trigger:**

* Extended from Manage Room.

**5.6. Use case: Manage Booking**

**5.6.1. Description:** Administrator can view and manage customer bookings.

Includes:

* View Booking History

**5.7. Use case: View Booking Customers’ Histories**

**5.7.1. Description:** Administrator can see the booking records of customers.  
**5.7.2. Trigger:**

* Extended from Manage Booking.

**5.8. Use case: View Reports**

**5.8.1. Description:** Administrator can generate and view reports about room usage, bookings, and payments.

**5.8.2. Postconditions:**

* Reports are displayed or exported for analysis.

**V. Class diagram:**

A diagram of a computer

AI-generated content may be incorrect.

**VI. Data model:**

A diagram of a computer

AI-generated content may be incorrect.

**BẢNG ĐÁNH GIÁ THÀNH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **MSSV** | **Nhiệm vụ** | **Mức độ hoàn thành** |
| Phan Việt Hoài | 054206003120 | Đặc tả Func, Non-Func, class diagram | 100% |
| Trịnh Thị Ngọc Điệp | 051306006340 | Đặc tả Func, Non-Func, DFD | 100% |
| Phạm Quỳnh Trang | 034306001451 | Đặc tả Func, Non-Func, Use-case | 100% |
| Bùi Thị Minh Trang | 093306005542 | Đặc tả Func, Non-Func, Data model | 100% |