1. Use case Login

|  |  |  |
| --- | --- | --- |
| Use Case Name | Login | |
| Scenario: | User attempts to log into the system | |
| Triggering Event: | The user clicks the “Login” button on the system’s main interface | |
| Brief Description: | User enters username and password. System verifies credentials. If correct, access is granted | |
| Actors: | * User | |
| Related Use Cases: | * Sign up * Post a review * Manage staff list * Manage room list * Clean assigned room * Manage personal information * Manage customer reservation * Manage guest parking | |
| Stakeholders: | All user types: require secure and accurate access to the system | |
| Preconditions: | * The user must have an existing account * The user has valid credentials (username and password) * The system is online and accessible | |
| Postconditions: | * If authentication is successful, the user is granted access to the system based on their role * The user session is created and maintained * If login fails, the user remains on the login page and is prompted to try again | |
| Flow of Events: | Actor System | |
| 1. User opens login page 2. User enters username and password 3. User clicks “Login” button | * 1. System displays the login interface   2.1. System validates credentials   * 1. If credentials are correct, system redirects to user appropriate page   2. If credentials are incorrect, system shows error message |
| Exception Conditions: | 2.2. If user leaves fields empty, system displays a warning and prevents login  2.3. If user account is locked or disabled, system displays account status message  3.3. If database is unreachable, system shows “Try again later” message | |

1. Sign up

|  |  |  |
| --- | --- | --- |
| Use Case Name | Sign up | |
| Scenario: | A new customer registers for an account on the hotel management system | |
| Triggering Event: | Customer clicks on the “Sign up” button on the system’s interface | |
| Brief Description: | Customer provides required information such as name, email, and password. The system validates the input and creates a new customer account | |
| Actors: | * Customer | |
| Related Use Cases: | * Login * Manage personal information * Make a reservation * Post a review | |
| Stakeholders: | Customer: wants a fast and easy registration process | |
| Preconditions: | * The customer is not already logged in * The customer does not have an existing account with the same email | |
| Postconditions: | * A new user account is created and saved in the system * The customer can now log in using their credentials | |
| Flow of Events: | Actor System | |
| 1. Clicks “sign up” button 2. Fills in required fields 3. Submits registration | * 1. Displays sign up form   3.1. Validates input  3.2. Creates account and store user information |
| Exception Conditions: | 3.3. If email is already in use, system notifies the customer  3.4. If required fields are missing or invalid, system prompts for correction  3.5. If there is a server or database error, account creation fails and the customer is informed | |

1. Make a reservation

|  |  |  |
| --- | --- | --- |
| Use Case Name | Make a reservation | |
| Scenario: | A customer books a room through the hotel management system | |
| Triggering Event: | Customer selects the reservation option and provides necessary booking details | |
| Brief Description: | The system collects reservation details then verifies availability and confirms the booking | |
| Actors: | * Customer | |
| Related Use Cases: | * Login * Sign up * Request for support * Post a review | |
| Stakeholders: | * Customer: expects a smooth booking process and accurate availability * Receptionist: manage or modify the reservation * Administrator: wants optimized room usage and accurate records | |
| Preconditions: | * Rooms must exist in the system with availability status | |
| Postconditions: | * A reservation record is created and linked to the customer * Room availability is updated accordingly * The customer receives booking confirmation | |
| Flow of Events: | Actor System | |
| 1. Inputs check-in/check-out, quantity of rooms, adults, children  2. Clicks “Check availability” | * 1. Display booking form   2.1. System validates form inputs  2.2. System check room availability  2.3. If available, provide a list of available rooms  2.4. Process payment and send successful booking notification |
| Exception Conditions: | 1.1. If required fields are empty or invalid, system displays validation errors  1.2. If no rooms are available for the selected criteria, system informs the customer and suggests alternatives  2.5. If the system encounters an error while saving the reservation, the booking fail and the customer is notified | |

1. Request for support

|  |  |  |
| --- | --- | --- |
| Use Case Name | Request for support | |
| Scenario: | Customer clicks the support icon to copy contact information for assistance | |
| Triggering Event: | Customer clicks on the support icon (e.g., phone, email, or social media) from the system interface | |
| Brief Description: | Upon clicking an icon, the corresponding contact detail is copied to the clipboard for the customer to use in their preferred communication tool | |
| Actors: | * Customer | |
| Related Use Cases: | None | |
| Stakeholders: | * Customer: needs a fast way to get support information * Hotel staff: expects contact requests via provided channels | |
| Preconditions: | * Customer accesses the system interface (no login required) * Support icons and information are visible and functional | |
| Postconditions: | * The selected contact information is copied to the customer’s clipboard * The customer is optionally shown a success message | |
| Flow of Events: | Actor System | |
| 1. Clicks on a support icon 2. Describes issue | * 1. Detects the selected contact method   2. Copies contact information to clipboard   3. Display message “Copy to clipboard” |
| Exception Conditions: | 1.2: If clipboard access is denied or fails, the system displays an error message and suggests manual copy  1.3: If the icon/link is broken or unavailable, the system logs the error or shows fallback contact info | |

1. Post a review

|  |  |  |
| --- | --- | --- |
| Use Case Name | Post a review | |
| Scenario: | A customer who has completed a reservation submits a review about their hotel stay | |
| Triggering Event: | Customer selects the option to post a review after completing a reservation | |
| Brief Description: | The system stores the review, and it can be viewed by administrators for reference, but not edited or deleted. Reviews are not publicly visible to customers | |
| Actors: | * Customer * Admin | |
| Related Use Cases: | * Make a reservation | |
| Stakeholders: | * Customer: wants to share feedback after using the service * Admin: wants to collect feedback for service quality assessment | |
| Preconditions: | * Customer has completed a reservation | |
| Postconditions: | * The review is saved in the system and becomes available to administrators * The review cannot be edited or deleted | |
| Flow of Events: | Actor System | |
| 1. Customer accesses review page 2. Writes and submits review form | * 1. Verifies customer has a completed reservation   2. Validates and saves the review   2.1. Displays confirmation message |
| Exception Conditions: | 1.2: If saving fails (e.g., system/database error), notify the customer | |

1. Manage personal information

|  |  |  |
| --- | --- | --- |
| Use Case Name | Manage personal information | |
| Scenario: | A registered user logs in and updates or views their personal profile information | |
| Triggering Event: | User selects the option to view or update their personal information after logging in | |
| Brief Description: | User views and edits their profile, such as name, phone or email | |
| Actors: | * User | |
| Related Use Cases: | * Login * Sign up | |
| Stakeholders: | User: want to keep their information accurate and updated | |
| Preconditions: | * The user has an existing account * The user is logged into the system | |
| Postconditions: | * Any updates made to personal information are saved and reflected in the system | |
| Flow of Events: | Actor System | |
| 1. Logs into the system 2. Navigates to "Personal Information" section 3. Makes edits and submits changes | * 1. Authenticates user credentials   2.1. Displays current user information  3.1. Validates the input data  3.2. Saves changes and confirms success |
| Exception Conditions: | 1.1: If login fails, access to personal information is denied  3.1: If required fields are missing or input is invalid, display error messages  3.2: If saving fails (e.g., system/database issue), notify the user | |

1. Manage customer reservation

|  |  |  |
| --- | --- | --- |
| Use Case Name | Manage customer reservation | |
| Scenario: | The receptionist handles customer reservation requests, including viewing, adding, updating, or canceling bookings | |
| Triggering Event: | The receptionist logs into the system and accesses the reservation management section | |
| Brief Description: | The receptionist views, adds, updates, or cancels customer reservations | |
| Actors: | * Receptionist | |
| Related Use Cases: | * Make a reservation * Login | |
| Stakeholders: | * Receptionist: wants to manage customer bookings accurately * Customer: expects correct handling of their reservations | |
| Preconditions: | * Receptionist is authenticated (logged in). | |
| Postconditions: | * Reservation status or details are updated accordingly in the system | |
| Flow of Events: | Actor System | |
| 1. Logs into the system 2. Opens reservation management section 3. Selects to add, edit, or delete a reservation | * 1. Authenticates receptionist credentials   2.1. Displays list of current reservations  3.1. Processes the request and updates reservation status or details  3.2. Saves changes and displays confirmation |
| Exception Conditions: | * 1.1: If login fails, access is denied * 3.2: If update fails (e.g., database error), display error message and suggest retrying the update | |

1. Manage room list

|  |  |  |
| --- | --- | --- |
| Use Case Name | Manage room list | |
| Scenario: | The administrator accesses the room management interface to add, update, or remove room records in the system | |
| Triggering Event: | Administrator logs into the system and navigates to the manage room list section | |
| Brief Description: | The administrator adds, updates, or removes rooms in the system | |
| Actors: | * Administrator | |
| Related Use Cases: | * Login | |
| Stakeholders: | * Administrator: needs to maintain an accurate and updated list of rooms * Receptionist: depends on accurate room data to manage reservations * Customers: indirectly affected by how rooms are categorized and offered | |
| Preconditions: | * Administrator is authenticated (logged in). * The system has access to the room database. | |
| Postconditions: | Room list is updated in the system | |
| Flow of Events: | Actor System | |
| 1. Logs into the system 2. Opens room management section 3. Selects to add, edit, or delete a room | 1.1. Verifies administrator credentials  2.1. Displays current room list  3.1. Validates input and processes the request  3.2. Updates the room database and confirms the action |
| Exception Conditions: | * 1.1: If login fails, access is denied * 3.1: If input is invalid, system displays error * 3.2: If database update fails, notify the user and suggest retrying | |

1. Manage staff list

|  |  |  |
| --- | --- | --- |
| Use Case Name | Manage staff list | |
| Scenario: | The administrator accesses the staff management interface to add, update, or remove staff information records in the system | |
| Triggering Event: | The administrator logs into the system and navigates to the manage staff list section | |
| Brief Description: | The administrator adds, updates, or removes staff accounts and roles in the system | |
| Actors: | * Administrator | |
| Related Use Cases: | * Login | |
| Stakeholders: | * Administrator: responsible for managing personnel records * Staff: depend on accurate information for role-based access * System: uses this data to authorize and route responsibilities | |
| Preconditions: | * Administrator is logged into the system * Staff data is stored in an accessible database | |
| Postconditions: | Staff list is updated in the system | |
| Flow of Events: | Actor System | |
| 1. Logs into the system 2. Opens staff management section 3. Selects to add, edit, or delete a staff | 1.1. Verifies administrator credentials  2.1. Displays current staff list  3.1. Validates input and processes the request  3.2. Updates the staff database and confirms the action |
| Exception Conditions: | * 1.1: If login fails, access is denied * 3.1: If input is invalid, system displays error * 3.2: If database update fails, notify the user and suggest retrying | |

1. Clean assigned room

|  |  |  |
| --- | --- | --- |
| Use Case Name | Cleaned assigned room | |
| Scenario: | The housekeeper logs into the system to view and update the status of their assigned rooms after cleaning tasks are completed | |
| Triggering Event: | The housekeeper accesses the system and navigates to the assigned room cleaning task list | |
| Brief Description: | The housekeeper views a list of rooms assigned to them and update the cleaning status of each. It ensures the hotel staff are informed of room readiness for the next customer | |
| Actors: | * Housekeeper | |
| Related Use Cases: | * Login | |
| Stakeholders: | * Housekeeper: uses the system to organize and report task progress * Receptionist: relies on updated room status to assign rooms to customers * Administrator: monitors overall housekeeping performance and workload | |
| Preconditions: | * Housekeeper is authenticated and has assigned cleaning tasks * Room status data is available in the system | |
| Postconditions: | * Room cleaning status is updated in the system | |
| Flow of Events: | Actor System | |
| 1. Logs into the system 2. Opens cleaning service management section 3. Selects to add, edit, or delete a cleaning service | 1.1. Verifies housekeeper credentials  2.1. Displays room list with current cleaning status  3.1. Validates input and processes the request  3.2. Updates the cleaning service database and confirms the action |
| Exception Conditions: | * 1.1: If login fails, access is denied * 3.1: If input is invalid, system displays error * 3.2: If database update fails, notify the user and suggest retrying | |

1. Manage guest parking

|  |  |  |
| --- | --- | --- |
| Use Case Name | Manage guest parking | |
| Scenario: | The parking staff logs into the system to view, add, or update guest parking information during check-in and check-out processes | |
| Triggering Event: | The parking staff logs in and selects parking management | |
| Brief Description: | The parking staff registers parking details, update parking status, and manage the guest's parking record during their stay | |
| Actors: | * Parking staff | |
| Related Use Cases: | * Login * Make a reservation | |
| Stakeholders: | * Parking Staff: maintains accurate parking records * Guests: expect secure and well-tracked parking services * Administrator: monitors overall parking operations | |
| Preconditions: | * Parking staff is authenticated * Guest reservation exists in the system * Parking data storage is available | |
| Postconditions: | * Parking information is successfully added or updated | |
| Flow of Events: | Actor System | |
| 1. Logs into the system 2. Opens guest parking management section 3. Selects to add, edit, or delete a guest parking information | 1.1. Verifies parking staff credentials  2.1. Displays list of current guest parking information  3.1. Validates input and processes the request  3.2. Updates the guest parking information database and confirms the action |
| Exception Conditions: | * 1.1: If login fails, access is denied * 3.1: If input is invalid, system displays error * 3.2: If database update fails, notify the user and suggest retrying | |