Faculty of Engineering and Architecture
Department of Computer Engineering
CEN 302-Software Engineering

Hotel Management System Requirements Specification

Version 1.0

March 31st, 2025

Prepared by:

Laert Mema

Alteo Kadriu

Danjel Ramku

Endri Demaj

Enrald Kadiu

Jurgen Balliu

Luis Kadiu

Received by:

M.Sc Shkumbin Fida

M.Sc Edlira Cani

Table of Contents

1) Executive Summary

a) Project Overview

The Hotel Management System (HMS) is a comprehensive web-based solution designed to modernize the way hotels handle room reservations, guest check-ins and check-outs, and overall operational management. Currently, many hotels still rely on manual, paper-based processes for managing bookings, guest records, and staff coordination. These traditional methods are not only time-consuming and prone to errors but also lack the efficiency and accessibility required by modern hospitality businesses and their customers.

HMS aims to address these challenges by providing a centralized, digital platform that streamlines and automates various aspects of hotel operations, from room reservations and guest management to staff administration and financial reporting. By leveraging cutting-edge web technologies, HMS offers a user friendly interface accessible from any device with an internet connection, enabling seamless collaboration and coordination among hotel guests, receptionists, housekeeping staff, and hotel managers. This system enhances operational efficiency, improves customer experience, and ensures a smooth and hassle-free hotel management process.

b) Purpose and Scope of this Specification

The purpose of this project is to develop a web application that will facilitate the operations of hotels by digitalizing and automating their management processes. This application aims to replace traditional, paper-based booking and record-keeping systems, allowing for easier access, better management, and enhanced security of hotel data. The system will enable guests to make reservations online, streamline the check-in and check-out process for receptionists, and simplify administrative tasks for hotel managers and staff. This documentation provides detailed information about the functionalities and features of the software, catering to all users, including hotel guests, receptionists, housekeeping staff, and management.

2) Product/Service Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

a) Product Context

Our software is designed for hotels and resorts nationwide, serving as a comprehensive and standalone system accessible to managers, receptionists, guests, and cleaners. It streamlines hotel operations, enhances guest experiences, and optimizes communication and task management among staff. The system ensures seamless coordination between departments—such as housekeeping, the front desk, and management—facilitating smooth daily operations and exceptional service delivery.

b) User Characteristics

- 1. Manager:
- Generate detailed reports on hotel occupancy rates, revenue, guest feedback, and staff performance.
- Manage and update staff information, including hiring new employees or modifying existing records.
- Maintain and update the hotel's service offerings, such as room types, amenities, and pricing, ensuring accurate and up-to-date information for guests.
- Track and manage the hotel's inventory, including housekeeping supplies, linens, and equipment, ensuring adequate stock levels.

- Oversee and coordinate the overall operations of the hotel, ensuring smooth functioning and adherence to hospitality standards.

2. Receptionist:

- Create and manage room reservations for new and returning guests, capturing all necessary guest information.
- View and manage a centralized list of room bookings, check-ins, and check-outs for all guests.
- Modify or cancel reservations based on guest requests or operational requirements.
- Assist guests with inquiries, provide information about hotel services, and address any concerns during their stay.
- Process check-ins and check-outs efficiently, ensuring a seamless guest experience.
-

3. Guest:

- View available room types, amenities, and pricing to make informed booking decisions.
- Create, modify, or cancel reservations online or through the hotel's front desk.
- Access reservation details, including check-in/check-out times, room preferences, and special requests.
- Provide feedback or reviews about their stay to help improve hotel services.
- Request additional services, such as room service, housekeeping, or concierge assistance, during their stay.
-

4. Cleaner (Housekeeping Staff):

- View a list of rooms assigned for cleaning, including check-out rooms and stay-over rooms requiring service.
- Update the status of rooms (e.g., cleaned, inspected, or maintenance required) in the system after completing tasks.
- Report any maintenance issues or damages observed in guest rooms to the management.
- Ensure rooms are stocked with necessary supplies, such as toiletries, towels, and linens.
- Coordinate with the reception team to prioritize cleaning tasks based on guest arrivals and departures.

c) Assumptions

• Receptionists, housekeepers, and the hotel manager will receive proper training to use the application efficiently.

• The hotel has the necessary IT infrastructure (computers, tablets, internet connectivity) to support the implementation and smooth operation of the system.

• Proper verification of guest details and reservation data will be conducted to maintain data integrity.

• Guests will use the application to book rooms, request services, and provide feedback without external assistance.

• Receptionists are responsible for ensuring that reservations are recorded accurately and that check-ins/check-outs are processed correctly.

• Housekeepers will update room statuses (cleaned, requires maintenance) in a timely and accurate manner.

• The hotel manager is responsible for maintaining up-to-date records of hotel staff, available rooms, and services within the system.

d) Constraints

- • The Hotel Management System requires a stable internet connection to function properly, as it is a web-based solution.
- • The system's efficiency depends on the staff's proper training and adoption of the application.
- • The performance and reliability of the application are dependent on the hotel's IT infrastructure, including network stability and device availability.
- • The accuracy and completeness of guest and room data impact the effectiveness of search functionalities (US_04, US_11).
- • Features like invoice generation and room key management rely on proper configuration and integration within the application (US_02, US_03).
- • The application's user experience and functionality may be constrained by the limitations of the web technologies and frameworks used in development.
- • The ability to generate reports and track minibar usage depends on accurate data entry by housekeepers and receptionists (US_14).

e) Dependencies

- • The creation of new reservations depends on the availability and active participation of receptionists (US_01, US_05).
- • The check-in and check-out processes rely on guest availability and proper data handling by receptionists (US_02, US_03).
- • Housekeeping tasks, such as marking rooms as cleaned and reporting issues, are dependent on housekeepers updating the system (US_10, US_12).
- • Maintenance requests depend on housekeepers reporting issues and the hotel's maintenance team responding accordingly (US_13).
- • Room service and housekeeping requests depend on timely communication between guests, receptionists, and hotel staff (US_06, US_07).
- • The ability to view and manage hotel services depends on the manager ensuring that information is updated and accurate in the system (US_09).
- • Guest feedback collection and review management are dependent on guests submitting feedback through the system (US_08).

3) Requirements

a) Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
Req#	Requirement	Comment	Priority	Date	Reviewed/ Approved
FR_01	The application should have a different view for different user roles.	Every role will have its own dashboard.	2	25/03/2025	Danjel Ramku/ Laert Mema

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
FR_02	The user's passwords must be secure	The passwords will be saved in the database as hashes instead of plain text	1	25/03/2025	Endri Demaj/ Laert Mema
FR_03	User will be able to change his password, but not the username	The username will be generated automatically by combining first name and last name	1	25/03/2025	Alteo Kadriu/ Jurgen Balliu
FR_04	The guest should be able to view all available room types	It is crucial for the guest to choose which room to book.	1	25/03/2025	Laert Mema/ Enrald Kadiu
FR_05	The guest should be able to make a new reservation, modify or cancel it.	The guest enters the details required for the reservation and can modify them or cancel the reservation	1	25/03/2025	Luis Kadiu/ Endri Demaj
FR_06	The guest should be able to order for extra amenities and services	The guest is able to request additional services to his reservation	2	25/03/2025	Alteo Kadriu/ Laert Mema
FR_07	The guest should be able to report issues with the room and ask for maintenance	Guests can report issues related to their stay and seek support from staff.	2	25/03/2025	Alteo Kadriu/ Laert Mema*
FR_08	The guest makes a payment for the reservation	A payment must be made for the reservation as it is legally complied.	1	25/03/2025	Danjel Ramku/ Luis Kadiu
FR_09	The receptionist handles check-in/check-out of the users.	Receptionists manage guest check-ins and check-outs, updating room status accordingly.	1	25/03/2025	Endri Demaj/ Jurgen Balliu
FR_10	The receptionist can view, update and manage all reservations	Important feature for reservation oversight by receptionists	1	25/03/2025	Enrald Kadiu/ Endri Demaj
FR_11	The receptionist can access and update guest information	Receptionist can manage guest profiles and booking details	1	25/03/2025	Luis Kadiu/ Laert Mema
FR_12	The receptionist can notify housekeeping	The receptionist can notify housekeeping or facilitate other services	2	25/03/2025	Endri Demaj/ Danjel Ramku

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
FR_13	The housekeeper updates the status of the room after cleaning	After cleaning the housekeeper changes the status of the room to clean	2	25/03/2025	Jurgen Balliu/ Alteo Kadriu
FR_14	The housekeeper can see which rooms are uncleaned	This helps the housekeeper with the cleaning schedule and task assignment	2	25/03/2025	Danjel Ramku/ Laert Mema
FR_15	The manager is able to access staff, guest and reservation lists	The manager has access to every list	2	25/03/2025	Laert Mema/ Danjel Ramku
FR_16	The manager can add or remove staff members	The manager is responsible for staff management	2	25/03/2025	Endri Demaj/ Luis Kadiu
FR_17	The manager can add new services	The manager can introduce new service packages	2	25/03/2025	Endri Demaj/ Jurgen Balliu
FR_18	The manager can update the prices per room type.	The manager is responsible for the prices charged per room	1	25/03/2025	Enrald Kadiu/ Laert Mema
FR_19	The manager can view and generate financial reports	The manager can view financial reports containing the revenue from the all the payments during a certain period	2	25/03/2025	Endri Demaj/ Luis Kadiu
FR_20	The manager can review customer feedback	Manager can read all the feedback by guests	3	25/03/2025	Jurgen Balliu/ Danjel Ramku
FR_21	The web application shall be responsive	The application should be accessible to every device	1	25/03/2025	Enrald Kadiu/ Luis Kadiu
FR_22	The application shall implement modern authentication and authorization.	To ensure the application confidentiality, these methods shall be applied	1	25/03/2025	Laert Mema/ Alteo Kadriu

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
FR_23	All the inputs received by the application will be validated	To protect against malicious attacks, every input will be validated thoroughly	1	25/03/2025	Laert Mema/ Endri Demaj

b) Non-Functional Requirements

i) Product Requirements

(1) User Interface Requirements

The user interface for the web application should be compatible with any browser in order for the user to access it from Desktop or Mobile. The user interface is divided into 6 main groups:

- Home interface
  Contains:
  - A navigation bar containing application name, buttons for logging in and registering and options such as Rooms, Amenities and Contact
  - The body will contain sliding images of the hotel's rooms and option to view details for each type of room
  - A reservation button, to click to reserve a room
  - Footer, which includes contact details, a short description and quick links to navigate through the application

- Login/Register interface:
  - The login and register interface will be similar. The register interface will contain fields : First Name, Last Name, Country and all the other required details for a user to register in the application.
  - The login interface will contain fields for first and last name and password. Also, most importantly the button for logging in.
  - All the inputs from these interfaces will be thoroughly validated, and if invalid an error will display in the screen, instructing the user to enter valid details
- Guest interface:

  - A navigation bar containing options the same as in the home interface : Amenities, Contact and Rooms.
  - A button to make new reservations, when clicked a reservation form will appear which will need details such as Check-In and Check-Out dates, Room Type, number of guests and options to include additional services. The form will include a textfield to write about special requests. In the end, the total summary and amount to be paid shows. The user also chooses a payment method and then proceeds with the reservation by clicking the make reservation button

- A table with all the upcoming, past and cancelled bookings by the user shows and their status which is “Confirmed” or “Cancelled”. Each upcoming booking has the options to view reservation details or cancel the reservation.
- Footer, which includes contact details, a short description and quick links to navigate through the applicati

- Receptionist interface:

  - Interface will contain a sidebar, containing options such as “Room Management”, “Guest Services”, “Reservations”, “Check-In/Check-Out”, “My Profile” and “Log Out”. In the main body there will be a dashboard overview which shows occupied rooms, free rooms, number of today’s arrivals and departures.
  - In the body of the dashboard, there will be a list of the reservations the receptionist will check-in for the day and another list of reservations that will be checked-out by the receptionist.
  - A “New Reservation” button which when clicked will allow the receptionist to fill a form with all the details required for a reservation.
  - When clicking “Reservations”, the receptionist will be able to see all the upcoming and past reservations.
  - When clicking “ Guest services”, a list of services required by every reservation will appear.
  - In the “Check-In/Check-Out”, the receptionist will click a reservation to check-in or out. The room status will be updated automatically. When checking-in, a list of available rooms of the selected reservation type will appear and the receptionist will assign that room to the reservation.
  - In the “Room Management” section, a list of available rooms will appear for every reservation type.
  - On top of the page, additional details such as the time and the date and a welcome message will appear.

- Housekeeper interface:

  - On top of the page, additional details such as the time and the date and a welcome message will appear.
  - Interface will contain a sidebar, containing options such as “Room Status”, “My tasks”, “Dashboard”, “My Profile” and “Log Out”. In the main body there will be a dashboard overview which shows cleaned rooms, number of tasks, tasks completed and rooms to clean.
  - In the main body of the dashboard there is a table which will shows the housekeeper the rooms that he will clean for the day
  - In the “My tasks” menu , a list with all the completed and pending tasks will appear and here the housekeeper can mark the task as complete, which will mean the room is cleaned and ready.
  - In the “Room Status” menu, a list of rooms will show according to their status “Dirty”, “Cleaning”, “Clean”.
  - The “My Profile” menu will show all the user details

- Manager interface:

  - On top of the page, additional details such as the time and the date and a welcome message will appear.
  - Interface will contain a sidebar, containing options such as “Staff Management”, “Overview”, “Reports”, “My Profile” and “Log Out”. In the main body there will be a dashboard overview which shows total rooms, occupancy rate, the daily generated revenue and pending reservations.
  - The main body of the dashboard will contain a list of recent reservations with their details such as Status and their Check-In/Check-Out dates.
  - In the “Reports” menu, the manager can select a date range and a report type and a report will be generated. The report can be of “Occupancy Report” or “Revenue Report”.
  -A list of generated reports in the last 30 days will appear.
  -In the “Staff Management” menu, a list of the entire staff will appear and their respective roles. The manager will be able to edit their details and roles. Also the manager can add or delete staff members. -To add staff members, a form appears with all the details required to add a new staff member.

(2) Learnability

The Hotel Management System (HMS) shall be designed to ensure that users can quickly and easily learn how to operate the system with minimal training. Key learnability features include:

- **Intuitive Interface**: The system will provide a clear, consistent, and user-friendly interface tailored to each user role (Guest, Receptionist, Housekeeper, Manager).
- **Guided Onboarding**: New users will receive tooltips, contextual help, or a brief tutorial upon first login to familiarize them with core functionalities.
- **Role-Based Dashboards**: Each user role will have a customized dashboard with only relevant functions, reducing complexity.
- **Consistent Navigation**: Common actions (e.g., reservations, check-ins) will follow standardized workflows across the system.
- **Minimal Training Dependency**: The system will rely on familiar design patterns (e.g., form submissions, button placements) to reduce the need for extensive training.

(3) Accessibility

The HMS shall adhere to accessibility standards to ensure usability for all users, including those with disabilities:

- Compliance: The web interface will follow Web Content Accessibility Guidelines (WCAG) 2.1, including proper contrast, keyboard navigation, and screen reader compatibility.
- Responsive Design: The system will be fully functional on desktops, tablets, and mobile devices, adapting to screen sizes.
- Text Alternatives: Images and icons will include alt text for screen readers.
- Adjustable Text: Users can zoom in or adjust font sizes without breaking the interface.

(4) Efficiency

The system shall optimize task completion time and resource usage:

- **Quick Actions**: Frequent tasks (e.g., check-ins, room status updates) will have shortcuts or batch-processing options.
- **Performance**: Pages will load within **2 seconds** under normal network conditions.
- **Search & Filters**: Users can quickly locate reservations, rooms, or guest details using dynamic search and filtering.
- **Automation**: Repetitive tasks (e.g., invoice generation, report scheduling) will be automated where possible.

(5) Dependability/Memorability

The HMS shall ensure users can easily recall how to perform tasks after periods of inactivity:

- **Consistent Layout**: Key features (e.g., reservation forms, room status updates) will maintain the same placement across sessions.
- **Visual Cues**: Icons and labels will be standardized (e.g., a "bed" icon always represents room management).
- **Saved Preferences**: User-specific settings (e.g., default date ranges for reports) will persist between logins.
- **Task History**: Recently accessed reservations or rooms will be highlighted for quick reference.
- **Error Recovery**: Clear instructions will guide users to correct mistakes (e.g., "Invalid date format: Use DD/MM/YY

(6) Errors

The system shall minimize user errors and provide clear recovery paths:

1. Preventive Design: Input validation will block invalid entries (e.g., future check-out dates before check-in).
2. Descriptive Messages: Errors will explain the issue and suggest fixes (e.g., "Room already booked. Please select another date").
3. Undo Actions: Critical actions (e.g., reservation cancellations) will include confirmation dialogs and optional undo periods.
4. Logging: All errors (user-facing and system) will be logged for troubleshooting, with no sensitive data exposed.
5. Graceful Degradation: If a feature fails (e.g., payment processing), the system will retain progress and guide users to alternative steps.

3.2.1.7 Satisfaction

The Hotel Management System is designed to be simple and easy to use for everyone, from receptionists to managers and even guests. The interface is intuitive, so users can quickly find what they need without spending too much time learning the system.

Everything is set up to make daily tasks smoother, whether it’s checking in guests, managing bookings, or handling payments. Plus, since the system works on both computers and mobile devices, users can access it anytime, making their jobs even easier.

Overall, the goal is to create a hassle-free experience that helps hotel staff work efficiently and keeps guests happy.

3.2.1.8 Capacity

The Hotel Management System will be developed to meet all the operational needs of a hotel. The system will function simultaneously for guests, receptionists, managers, and housekeeping staff. It will operate in real-time, ensuring that any updates or changes are immediately reflected for all authorized users.

All users will interact with the same database. If multiple requests are made to the server, they will be processed in a queue, which might cause slight delays. To improve performance and reduce server load, changes made by users will first be stored locally on their devices before being synchronized with the database. This approach minimizes disruptions and allows users to continue their tasks even during minor delays.

The application will be hosted on a web server. The system itself is lightweight, and the database storage requirements are minimal, ensuring efficient performance.

(7) Security

The data stored in the system's database is considered sensitive. Therefore, a high level of security must be ensured. In compliance with Law No. 9887, dated 10.03.2008, amended by Law No. 48/2012, "On the Protection of Personal Data," guest and staff data will be protected and accessible only to authorized personnel. This will be achieved through:

• Storing user passwords in an encrypted (hashed) format, preventing plain-text storage.

• Implementing modern authentication and authorization to ensure secure access for authorized users only.

• Restricting access based on different roles, where receptionists, managers, and staff members can only access relevant data.

• Using secure protocols for data transmission.

ii) Organizational Requirements

Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc

(1) Availability

The system should be available **24/7** to support guest reservations and hotel operations.

Regular backups should ensure data integrity and prevent data loss.

(2) Latency

Response time for critical operations (e.g., room booking, check-in, payment processing) should be **under 2 seconds**.

(3) Monitoring

The system should include a **dashboard for monitoring system performance**, including uptime, user activity, and booking trends.

Administrators should receive alerts for **system failures, unauthorized access, or unusual activity**.

(4) Maintenance

Regular software updates should be scheduled to ensure security and performance improvements.

System logs should be maintained to track user activity and troubleshoot issues.

(5) Operations

The system is designed to be intuitive, ensuring ease of use.
Users returning after a long period can quickly re-establish proficiency.
The interface includes icons, buttons, and descriptions to enhance usability.

(6) Standards Compliance

Our system is a newly developed platform designed to digitize hotel reservation and service management. Although it is a new system, it must comply with some existing standards. Guest information must include all required fields according to hotel management best practices. Additionally, identification standards and reservation data will follow industry regulations. Every operation will be in accordance with data protection laws and standard hotel procedures.

(7) Portability

• The system will be a web-based application, ensuring it operates consistently across all operating systems.
• A computer or mobile phone with an internet connection is all that is required to use the system.
• The system will be accessible from the most commonly used browsers and will provide an optimal user experience across different devices.

iii) External Requirements

– Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

(1) Protection

To protect the system from unauthorized access, malicious modifications, disclosure, destruction, or misuse, the following measures will be taken:
• Encrypting the most sensitive data, such as passwords, using hashing methods to maintain privacy.
• Logging and auditing user activities to ensure accountability in case of any issues or misuse.
• The receptionist is responsible for the accuracy of guest data they enter, and the system does not hold liability for incorrect data input.
• Validating all input data to prevent special characters and potential cyberattacks.
• Requiring user confirmation through pop-up windows for critical actions to avoid accidental mistakes.
• Each guest will only be able to view their personal information and related reservations.
• Receptionists and staff will have limited access based on their roles to ensure the privacy and security of guest data.

(2) Authorization and Authentication

The Authorization and Authentication factors:

- User authentication will be using username, password, and JWT to enhance security.
- Authorization will be based on user roles, ensuring that each user can access only the information and functionalities assigned to their role.
- Sessions will be used to manage the currently logged-in user.
- If the user tries to log in with the wrong credentials, an error message will be displayed.

(3) Legislative Requirements

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values

c) Domain Requirements

This Web Application operates in the field of Hotel Management. The main purpose is to digitize and automate hotel operations, making reservation management, guest services, and staff coordination more efficient.

However, data security and access control are of utmost importance. Sensitive information such as guest details, payment records, and staff data must be protected and accessible only to authorized users. The application will ensure that each user has role-based access, allowing them to interact only with the relevant features.

This system is intended for use within a specific hotel and does not need to integrate with external systems. The web application will be responsive, ensuring accessibility across different devices, and it will implement modern authentication and authorization methods to guarantee data confidentiality and secure transactions.

4) User Scenarios/Use Cases

4.1 Requirements Analysis

4.1.1 User Scenarios List

Nr	Name	Description
US_01	User logs in	User : managers,guests,receptionists and cleaners log in using username and password
US_02	Make a reservation	Guest: The guest makes a booking by selecting room type, check-in, and check-out dates.
US_03	Order extra services	Guest: Description: The guest orders extra services during their stay.
US_04	Leave Feedback	Guest: The guest can leave a feedback regarding the stay in the hotel.
US_05	View reservation	Guest: Guest can view his current and past reservations
US_06	Cancel reservation	Guest: Guest cancels a pending reservation
US_07	Check-In guest	Receptionist: The receptionist checks in a guest and assigns a room.
US_08	Check-Out guest	Receptionist: The receptionist checks out a guest and marks the room for cleaning.
US_09	Add new staff member	Manager: The manager adds a new staff member by clicking the “Add Staff” button and by filling out a form.
US_10	Delete a staff member	Manager: The manager removes a staff member
US_11	View reports	Manager: Can view the generated occupancy reports
US_12	View feedback	Manager: Can view all the left feedback by the users
US_13	Create/add new service	Manager: Can create a new service for the users
US_14	Manage room and service prices	Manager : Is able to view and update the prices based on demand
US_15	View all incoming/checked-out reservations	Receptionist: Can see all the reservations to be checked-in and out.
US_16	Report a issue	Guest: Can report an issue and a maintenance task for the staff is created.
US_17	Update cleaning status	Housekeeper: Updates the cleaning status of a room after completing the cleaning.
US_18	View Cleaning Tasks	Housekeeper:Views the assigned cleaning tasks
US_19	Generate Invoice	System: The system generates an invoice for the reservation
US_20	View available room types	Guest: Checks available room types at check in-check out dates
US_21	Processing payment	System: The system processes a payment made by

		the guest to an invoice
US_22	View maintenance request	Manager, Receptionist, Housekeeper: Can view the maintenance request section.
US_23	Complete maintenance request / feedback	Manager,Housekeeper: Can mark the maintenance request as completed.
US_24	Update reservation	Guest,Receptionist: Can make changes to his reservation such as order services, change the check-in date or add a special request
US_25	Manage rooms	Manager: Add, modify or remove rooms

4.1.2 User Scenarios Extended

US_01: User logs in:

1. User navigates to login page
2. User enters username and password
3. System validates credentials
4. System grants access according to user role
5. If credentials are invalid: System displays error message and prompts user to try again
6. In case of forgotten password: User requests password reset

US_02: Make a reservation:

1. User selects room type
2. User enters check-in and check-out dates
3. System checks availability
4. User provides guest details
5. System creates reservation
6. System generates invoice

US_03: Order extra services

1. User selects reservation
2. User browses available services
3. User adds services to order with quantity
4. System calculates total price
5. User confirms order
6. System creates service order

US_04: Leave feedback

1. Guest navigates to the feedback section.
2. Guest fills in feedback form.
3. Guest submits feedback.
4. System stores feedback.
5. System displays feedback submission confirmation to the guest.

US_05:View Reservations

1. Guest navigates to the reservations page.
2. System retrieves guest's reservations.
3. System displays reservations to the guest.

US_06: Cancel reservation

1. User selects reservation to cancel
2. User provides cancellation reason
3. System cancels reservation

4. System updates invoice status

US_07: Check-In Guest
1. Receptionist searches for reservation at “View Reservations”
2. System displays reservation details
3. Receptionist selects a specific room of the reserved type and assigns it to the reservation.
4. Receptionist completes check-in
5. System updates reservation and room status

US_08: Check-Out Guest
1. Receptionist clicks “View Reservations”
2. Receptionist selects checked-in reservation
3. System displays balance due and service charges
4. Receptionist processes payment
5. System completes check-out
6. System creates cleaning task for room

US_09: Add new staff member
1. Manager clicks on the “Manage Staff” section.
2. Manager clicks “Add Staff” button
3. Manager fills the form with the user details.
4. Manager clicks “Add”
5. System creates new user with the given credentials

US_10: Delete a staff member
1. Manager clicks on the “Manage Staff” section
2. Manager clicks on the staff member
3. Manager clicks “Delete” button
4. System removes user

US_11: View reports:
1. Manager selects report type
2. Manager enters date range or parameters
3. System processes data
4. System displays report with visualization

US_12: View feedback
1. Manager navigates to the feedback section.
2. System retrieves feedback.
3. System displays feedback to the manager.

US_13: Add new service
1. Manager clicks on “Services” section
2. Manager clicks “Add New” button
3. Manager fills the form with the price and name
4. Manager clicks “Add”
5. System adds new service

US_14: Manage room and service prices:
1. Manager clicks “Manage prices” section
2. Manager chooses the commodity that he wants to update
3. Manager chooses new price
4. Manager clicks “Update price”
5. System updates price

US_15: View all incoming/outgoing reservations
1. Receptionist logs in to the system
2. Receptionist navigates to reservation management section
3. Receptionist selects view type (Today's Arrivals or Today's Departures)
4. System retrieves and displays relevant reservations

5. Receptionist reviews reservation details
6. Receptionist can filter or sort the list if needed
7. Receptionist can select a specific reservation to view complete details
8. Receptionist can proceed to check-in or check-out process if needed

US_16: Report an issue
1. Guest experiences an issue during their stay
2. Guest navigates to support/help section in the system
3. Guest selects "Report Issue" option
4. System displays issue reporting form
5. Guest selects issue category (plumbing, electrical, cleanliness, etc.)
6. Guest provides detailed description of the issue
7. Guest indicates urgency level
8. Guest uploads photo of issue (optional)
9. Guest submits the issue report
10. System confirms receipt of the issue report
11. System creates a maintenance task for staff

US_17: Update cleaning status
1. Housekeeper logs in to the system
2. Housekeeper views assigned cleaning tasks
3. Housekeeper selects a room to clean
4. Housekeeper physically performs room cleaning
5. Housekeeper conducts room inspection
6. If maintenance issues are found, housekeeper notes them in the system
7. Housekeeper updates the room cleaning status in the system
8. Housekeeper adds any relevant cleaning notes
9. Housekeeper submits the status update
10. System changes room status from "Dirty" to "Clean"
11. System notifies reception of room availability

US_18: View Cleaning Tasks
1. Housekeeper logs in to the system
2. System automatically displays dashboard with assigned cleaning tasks
3. Housekeeper reviews task list showing room numbers, priorities, and statuses
4. Housekeeper can filter tasks by floor, priority, or status if needed
5. Housekeeper selects a specific task to view detailed information
6. Housekeeper reviews any special instructions for the selected task
7. Housekeeper can mark task as "In Progress" when starting
8. Housekeeper can navigate between tasks as work progresses

US_19: Generate Invoice
1. System detects trigger for invoice generation (reservation creation, service order, checkout)
2. System retrieves reservation information
3. System calculates room charges based on room type and duration
4. System adds any service charges to the invoice
5. System calculates subtotal amount
6. System applies appropriate tax rates
7. System calculates total amount due
8. System links invoice to the reservation
9. System updates invoice status (Pending or Due)
10. System makes invoice available for viewing by staff and guest

US_20: View available room types
1. Guest navigates to booking/reservation page
2. Guest enters desired check-in date
3. Guest enters desired check-out date
4. Guest specifies number of guests
5. Guest submits search request

6. System validates date inputs
7. System checks room availability for selected dates
8. System displays available room types with images, descriptions, and amenities
9. System shows pricing for each available room type
10. Guest reviews room options
11. Guest can select a room type to proceed with reservation

US_21: Processing payment

1. Payment process is initiated (by guest online or receptionist at checkout)
2. System retrieves invoice details
3. System displays total amount due
4. User selects payment method (credit card, cash, bank transfer, etc.)
5. If credit card payment:
   - User enters credit card details
   - System validates card information
   - System sends authorization request to payment processor
   - System receives authorization response
6. If cash payment:
   - Receptionist enters cash amount received
   - System calculates change if necessary
7. System records transaction details
8. System updates invoice status to "Paid"
9. System generates payment receipt
10. System sends receipt to guest email

US_22: View maintenance request

1. Staff member (Manager, Receptionist, Housekeeper) logs in
2. Staff navigates to maintenance request section
3. System retrieves maintenance requests based on staff role
4. System displays list of maintenance requests
5. Staff can apply filters based on status, priority, date, or location
6. Staff selects a specific request to view details
7. System displays detailed request information including history and current status
8. Staff reviews related room and guest information
9. If authorized, staff can update request status or add notes
10. Staff can navigate back to the request list

US_23: Complete maintenance request/feedback

1. Manager or Housekeeper navigates to maintenance requests
2. Staff member selects request to complete
3. Staff reviews request details
4. Staff physically performs required maintenance
5. Staff returns to the system
6. Staff selects "Mark as Completed" option
7. System prompts for completion notes
8. Staff enters details of work performed and any follow-up needed
9. Staff uploads photos of completed work (optional)
10. Staff submits completion form
11. System updates request status to "Completed"
12. System notifies relevant parties (guest, management)

US_24: Update reservation

1. Guest or Receptionist selects reservation to modify
2. System displays editable reservation information
3. User can update reservation dates if needed
   - System checks availability for new dates
   - System updates reservation dates if available
4. User can change room type if needed
   - System checks availability of new room type

   - System updates room type if available
5. User can add additional services
6. User can add or modify special requests
7. User confirms all reservation changes
8. System updates reservation details
9. System recalculates total price
10. System updates the invoice
11. System sends confirmation of changes to guest

US_25: Manage rooms
1. Manager navigates to room management section
2. System displays list of all rooms
3. For adding a new room:
   - Manager clicks "Add Room" button
   - System displays room creation form
   - Manager enters room details (number, type, floor, features)
   - Manager submits form
   - System creates new room
4. For modifying an existing room:
   - Manager selects room to modify
   - System displays room details form
   - Manager updates room information
   - Manager submits changes
   - System saves updated details
5. For removing a room:
   - Manager selects room to remove
   - System checks if room has active reservations
   - If no active reservations, system asks for confirmation
   - Manager confirms deletion
   - System removes room from inventory

4.1.3 User Cases

UC_01: User logs in

UC Name	User logs in
Dependency	None
Actors	Manager,Guest,Receptionist,Housekeeper
Preconditions	User has valid account in the system
Description of Main Sequence	1. User navigates to login page 2. User enters username and password 3. User clicks login button 4. System authenticates user 5. System identifies user role 6. System redirects to role-specific dashboard
Description of Alternative Sequence	**1. Invalid credentials:**System displays error message ;b. User re-enters credentials or requests password reset **2. Account locked**: a. System notifies user that account is locked b. User contacts administrator
Non-functional requirements	1. Login process should complete within 3 seconds 2. Password must be stored in encrypted format 3. System must log failed login attempts 4. System must lock account after 5 failed attempts
Postconditions	1. User is authenticated 2. User has access to role-specific features 3. System logs successful login

UC_02: Make a reservation

UC Name	Make a reservation
Dependency	US_01 (User logs in) for logged-in users.None for guest bookings
Actors	Guest
Preconditions	1. User is on reservation page 2. Room types are available in the system
Description of Main Sequence	1. Guest enters check-in and check-out dates 2. Guest enters number of guests 3. System checks availability 4. System displays available room types 5. Guest selects desired room type 6. Guest enters personal information 7. Guest provides any special requests 8. System calculates total price 9. Guest confirms reservation 10. System creates reservation 11. System generates invoice 12. System displays confirmation

Description of Alternative Sequence	**No availability:** ● System displays no availability message ● System suggests alternative dates **Guest abandons reservation:** ● Reservation not created
Non-functional requirements	1. Availability check should complete within 2 seconds 2. System must maintain data consistency during concurrent bookings 3. System must support multiple currencies 4. Confirmation email must be sent within 1 minute
Postconditions	1. New reservation is created in the system 2. Room availability is updated 3. Invoice is generated 4. Confirmation is sent to guest

UC_03: Order extra services

UC Name	Order extra services
Dependency	US_01 (User logs in) US_02 (Make a reservation)
Actors	Guest
Preconditions	1. Guest has an active reservation 2. Services are available in the system
Description of Main Sequence	1. Guest navigates to the services section. 2. System displays available services. 3. Guest selects desired services. 4. Guest specifies quantity/options. 5. Guest adds services to cart. 6. System displays service order summary. 7. System calculates total cost. 8. Guest confirms order. 9. System processes order. 10. System adds charges to the guest's invoice. 11. System sends confirmation.
Description of Alternative Sequence	**1. Service unavailable:** a. System notifies guest. **2. Guest cancels order:** a. Order is not processed.

Non-functional requirements	1. Service ordering must be available 24/7. 2. Order processing should complete within 3 seconds. 3. System must handle concurrent service orders. 4. Email confirmation with order details.
Postconditions	1. Service order is linked to the reservation. 2. Invoice is updated with service charges. 3. Relevant staff are notified of the service order.

UC_04: Leave feedback

UC Name	Leave feedback
Dependency	US_01 (User logs in) US_02 (Make a reservation)
Actors	Guest
Preconditions	1. Guest has completed a stay at the hotel. 2. Reservation status is "Completed."
Description of Main Sequence	1. Guest navigates to the feedback page. 2. System displays the feedback form. 3. Guest rates aspects of the stay. 4. Guest enters comments. 5. Guest submits feedback. 6. System validates input. 7. System saves feedback. 8. System sends a thank-you message. 9. System notifies the manager.
Description of Alternative Sequence	**1. Invalid input:** a. System highlights errors. b. Guest corrects errors. **2. Guest abandons feedback:** a. Feedback is not submitted.
Non-functional requirements	1. Feedback submission should be simple and intuitive. 2. The form should be mobile-friendly. 3. The comments field should accept at least 1000 characters.
Postconditions	1. Feedback is stored in the system. 2. Management is notified of new feedback. 3. Feedback is linked to the guest's reservation.

UC_05: View Reservation

UC Name	View Reservation
Dependency	US_01 (User logs in)

	US_02 (Make a reservation)
Actors	Guest, Receptionist
Preconditions	1. Guest is authenticated. 2. Guest has at least one reservation.
Description of Main Sequence	1. Guest navigates to "My Reservations." 2. System retrieves guest's reservations. 3. System displays the list of reservations. 4. Guest selects a specific reservation. 5. System displays detailed reservation information.
Description of Alternative Sequence	**1. No reservations:** a. System displays "No reservations found" message. b. System offers to create a new reservation.
Non-functional requirements	1. Reservation list should load within 2 seconds. 2. System should display reservations in chronological order. 3. Interface should clearly indicate reservation status. 4. Printable reservation details.
Postconditions	1. Guest can view reservation details. 2. Guest can navigate to modify or cancel the reservation if applicable.

UC_06: Cancel Reservation

UC Name	Cancel Reservation
Dependency	US_01 (User logs in) US_05 (View reservation)
Actors	Guest
Preconditions	1. Guest is authenticated. 2. Reservation exists with status "Reserved." 3. Cancellation is within the allowed timeframe.
Description of Main Sequence	1. Guest selects reservation to cancel. 2. Guest clicks "Cancel Reservation." 3. System displays cancellation policy. 4. System shows any applicable fees. 5. Guest confirms cancellation. 6. Guest provides cancellation reason. 7. System processes cancellation. 8. System updates reservation status. 9. System sends confirmation email.
Description of Alternative Sequence	**1. Cancellation not allowed:** a. System displays message explaining why. **2. Guest abandons cancellation:** a. Reservation remains active.
Non-functional requirements	1. Clear display of cancellation policy and fees. 2. Cancellation confirmation within 1 minute. 3. System must maintain audit trail of cancellations.

	4. Cancellation email must include all relevant details.
Postconditions	1. Reservation status changed to "Cancelled." 2. Room becomes available for booking. 3. Cancellation fee applied if applicable. 4. Refund processed if applicable.

UC_07: Check-In guest

UC Name	Check-In guest
Dependency	US_01 (User logs in) US_02 (Make a reservation)
Actors	Receptionist
Preconditions	1. Receptionist is authenticated. 2. Reservation exists with status "Reserved." 3. Guest has arrived at the hotel. 4. Current date is the check-in date.
Description of Main Sequence	1. Receptionist searches for reservation. 2. System displays reservation details. 3. Receptionist verifies guest identity. 4. System displays available rooms of reserved type. 5. Receptionist selects specific room. 6. Receptionist collects payment method. 7. Receptionist completes check-in process. 8. System updates reservation status. 9. System assigns room to guest. 10. System marks room as "Occupied." 11. Receptionist provides room key.
Description of Alternative Sequence	**1. No rooms available:** **a. Receptionist offers room upgrade.** **b. Guest accepts or declines.** **2. Guest arrives without reservation:** **a. Receptionist creates walk-in reservation.**
Non-functional requirements	1. Check-in process should be completed within 5 minutes. 2. System must ensure room is not double-booked. 3. Interface must clearly show room status and features. 4. Room key creation integrated with the system.
Postconditions	1. Reservation status changed to "CheckedIn." 2. Room status changed to "Occupied." 3. Guest has access to room. 4. System records check-in time.

UC_08: Check-Out Guest

UC Name	Check-Out guest
Dependency	US_01 (User logs in) US_07 (Check-In guest)

Actors	Receptionist
Preconditions	1. Receptionist is authenticated. 2. Reservation exists with status "CheckedIn." 3. Guest is ready to depart.
Description of Main Sequence	1. Receptionist searches for reservation. 2. System calculates final bill. 3. Receptionist reviews charges with guest. 4. Guest confirms charges. 5. Receptionist processes payment if needed. 6. Receptionist completes check-out. 7. System updates reservation status. 8. System marks room as "Dirty." 9. System creates cleaning task. 10. System emails receipt to guest. 11. Guest departs.
Description of Alternative Sequence	**1. Disputed charges:** a. Receptionist investigates. b. Receptionist adjusts bill if necessary. **2. Payment failure:** a. Try alternative payment method.
Non-functional requirements	1. Check-out process should be completed within 5 minutes. 2. System must calculate all charges accurately. 3. Invoice must be clearly formatted and itemized. 4. Receipt email must be sent immediately.
Postconditions	1. Reservation status changed to "Completed." 2. Room status changed to "Dirty." 3. Cleaning task created. 4. Invoice marked as "Paid." 5. Receipt provided to guest.

UC_09: Add new staff member

UC Name	Add new staff member
Dependency	US_01 (User logs in)
Actors	Manager
Preconditions	1. Manager is authenticated. 2. Manager has appropriate permissions.
Description of Main Sequence	1. Manager navigates to staff management. 2. Manager clicks "Add Staff." 3. System displays staff form. 4. Manager enters staff details. 5. Manager selects staff role. 6. Manager submits form. 7. System validates information. 8. System creates staff account. 9. System generates temporary password. 10. System sends credentials to staff email.

Description of Alternative Sequence	**1. Invalid information:** **a. System highlights errors.** **b. Manager corrects information.** **2. Duplicate email:** **a. System shows error.** **b. Manager uses different email.**
Non-functional requirements	1. Staff credentials must be securely transmitted. 2. Temporary password must meet complexity requirements. 3. Staff roles must have appropriate permission limits. 4. Email with credentials must be sent within 5 minutes.
Postconditions	1. New staff account created. 2. Staff account has appropriate role and permissions. 3. Staff receives login credentials. 4. Staff added to payroll system if integrated.

UC_10: Delete staff member

UC Name	Delete staff member
Dependency	US_01 (User logs in) US_09 (Add new staff member)
Actors	Manager
Preconditions	1. Manager is authenticated. 2. Manager has appropriate permissions. 3. Staff member exists in system.
Description of Main Sequence	1. Manager navigates to staff management. 2. Manager searches for staff member. 3. Manager selects staff member. 4. Manager clicks "Delete" button. 5. System displays confirmation dialog. 6. Manager confirms deletion. 7. System deactivates staff account. 8. System displays confirmation message.
Description of Alternative Sequence	**1. Staff has active assignments:** **a. System shows warning.** **b. Manager reassigns tasks.** **c. Manager continues with deletion.** **2. Manager cancels deletion:** **a. Staff account remains active.**
Non-functional requirements	1. System must maintain audit trail of staff deletions. 2. System must prevent accidental deletion with confirmation. 3. Deletion process should handle task reassignment. 4. Interface must clearly show active assignments.
Postconditions	1. Staff account deactivated (not typically permanently deleted). 2. Staff tasks reassigned if necessary. 3. Staff removed from schedules and assignments.

UC_11: View reports

UC Name	View reports
Dependency	US_01 (User logs in)
Actors	Manager
Preconditions	1. Manager is authenticated. 2. Manager has reporting permissions.
Description of Main Sequence	1. Manager navigates to reports section. 2. Manager selects report type. 3. Manager specifies date range. 4. Manager selects additional filters. 5. Manager clicks "Generate Report". 6. System processes data. 7. System displays report with visualizations.
Description of Alternative Sequence	**1. No data available:** **a. System displays "No data" message.** **2. Manager exports report:** **a. Manager selects export format.** **b. System generates file.** **c. System provides download link.**
Non-functional requirements	1. Reports should generate within 10 seconds. 2. Visualizations must be clear and accurate. 3. System must support various export formats (PDF, Excel, CSV). 4. Reports must maintain data confidentiality.
Postconditions	1. Report is displayed to manager. 2. Report can be exported if needed. 3. System logs report generation activity.

US_12: View feedback

UC Name	View feedback
Dependency	US_01 (User logs in) US_04 (Leave Feedback)
Actors	Manager
Preconditions	1. Manager is authenticated. 2. Feedback entries exist in the system.
Description of Main Sequence	1. Manager navigates to feedback section. 2. System displays feedback list. 3. Manager filters/sorts as needed. 4. Manager selects specific feedback. 5. System displays detailed feedback. 6. Manager reviews ratings and comments.
Description of Alternative Sequence	**1. No feedback entries:** **a. System displays "No feedback" message.**

	2. Manager responds to feedback: **a. Manager enters response.** **b. System saves response.** **c. System notifies guest if applicable.**
Non-functional requirements	1. Feedback list should load within 3 seconds. 2. System should support filtering by date, rating, and status. 3. Interface should highlight critical feedback. 4. System should support feedback analytics.
Postconditions	1. Manager can view feedback details. 2. Manager can respond to feedback if needed. 3. Manager can mark feedback for follow-up.

UC_13: Create/add new service

UC Name	Create/add new service
Dependency	US_01 (User logs in)
Actors	Manager
Preconditions	1. Manager is authenticated 2. Manager has service management permissions
Description of Main Sequence	1. Manager navigates to service management 2. Manager clicks "Add New Service" 3. Manager enters service details 4. Manager sets pricing information 5. Manager uploads service image 6. Manager specifies availability 7. Manager submits form 8. System validates information 9. System creates new service
Description of Alternative Sequence	1. Invalid information: a. System highlights errors b. Manager corrects information 2. Duplicate service name: a. System shows error b. Manager chooses different name
Non-functional requirements	1. Service creation should be completed quickly 2. Image upload should support common formats 3. Pricing must support different currencies 4. Service should be immediately available after creation

Postconditions	1. New service is created in system 2. Service is available for booking 3. Service appears in service catalog

UC_14: Manage room and service prices

UC Name	Manage room and service prices
Dependency	US_01 (User logs in)
Actors	Manager
Preconditions	1. Manager is authenticated 2. Manager has pricing management permissions 3. Rooms and services exist in system
Description of Main Sequence	1. Manager navigates to pricing management 2. Manager selects room type or service 3. Manager views current pricing 4. Manager enters new base price 5. Manager saves changes 6. System updates pricing 7. System applies new pricing to future bookings
Description of Alternative Sequence	1. Manager sets seasonal pricing: a. Manager specifies date range b. Manager enters special rate c. System adds seasonal pricing rule
Non-functional requirements	1. Price updates should be applied immediately 2. System must maintain price history 3. Interface must clearly show current and future pricing 4. System must support multiple currencies
Postconditions	1. Prices are updated in system 2. New prices apply to future bookings 3. System logs price change history

UC_15: View all incoming/checked-out reservations

UC Name	View all incoming/checked-out reservations
Dependency	US_01 (User logs in)
Actors	Receptionist
Preconditions	1. Receptionist is authenticated 2. Reservations exist in system
Description of Main Sequence	1. Receptionist logs in to system 2. Receptionist navigates to reservation management 3. Receptionist selects view type (Today's Arrivals or Today's Departures) 4. System retrieves relevant reservations 5. System displays reservation list 6. Receptionist reviews reservation details
Description of Alternative Sequence	1. No reservations for selected criteria: a. System displays "No reservations" message 2. Receptionist applies custom filter: a. Receptionist enters custom date range b. System filters accordingly
Non-functional requirements	1. Reservation list should load within 2 seconds 2. Interface must clearly indicate reservation status 3. List should be sortable by arrival/departure time 4. System should highlight VIP guests
Postconditions	1. Receptionist can view relevant reservations 2. Receptionist can select specific reservation for check-in/out 3. Receptionist is prepared for arriving/departing guests

UC_16: Report an issue

UC Name	Report an issue
Dependency	US_01 (User logs in), US_02 (Make a reservation)
Actors	Guest
Preconditions	1. Guest is authenticated 2. Guest has active reservation 3. Guest is currently staying at hotel

Description of Main Sequence	1. Guest navigates to support/help section 2. Guest selects "Report Issue" option 3. Guest selects issue category 4. Guest enters issue description 5. Guest indicates urgency level 6. Guest uploads photo (optional) 7. Guest submits report 8. System creates maintenance task 9. System notifies relevant staff
Description of Alternative Sequence	1. System error during submission: a. System saves draft b. Guest can retry submission
Non-functional requirements	1. Issue reporting should be available 24/7 2. Interface should be simple and mobile-friendly 3. Photo upload should support common formats 4. High-priority issues should trigger immediate notifications
Postconditions	1. Maintenance task created in system 2. Task assigned to appropriate staff 3. Guest receives confirmation 4. Maintenance staff notified of new task

UC_17: Update cleaning status

UC Name	Update cleaning status
Dependency	US_01 (User logs in) US_08 (Check-Out guest) or US_18 (View Cleaning Tasks)
Actors	Housekeeper
Preconditions	1. Housekeeper is authenticated 2. Room is assigned for cleaning 3. Room status is "Dirty"
Description of Main Sequence	1. Housekeeper logs in to system 2. Housekeeper views assigned tasks 3. Housekeeper selects room to clean 4. Housekeeper performs cleaning 5. Housekeeper inspects room 6. Housekeeper updates cleaning status 7. Housekeeper adds notes if needed 8. Housekeeper submits update 9. System changes room status to "Clean"
Description of Alternative Sequence	1. Maintenance issue found: a. Housekeeper reports issue b. System creates maintenance task c. Room status updated appropriately

Non-functional requirements	1. Mobile-friendly interface for on-the-go updates 2. Quick status update process (under 30 seconds) 3. System should work offline with sync when online 4. Real-time status updates to reception
Postconditions	1. Room status changed to "Clean" 2. Cleaning task marked as completed 3. Reception notified that room is ready 4. Maintenance task created if issues reported

UC_18: View Cleaning Tasks

UC Name	View Cleaning Tasks
Dependency	US_01 (User logs in)
Actors	Housekeeper
Preconditions	1. Housekeeper is authenticated 2. Cleaning tasks exist in system
Description of Main Sequence	1. Housekeeper logs in to system 2. System displays dashboard with tasks 3. Housekeeper reviews task list 4. Housekeeper can filter tasks if needed 5. Housekeeper selects specific task 6. System displays detailed task information 7. Housekeeper reviews instructions 8. Housekeeper can mark task as "In Progress"
Description of Alternative Sequence	1. No tasks assigned: a. System displays "No tasks" message 2. Housekeeper completes task: a. Redirects to Update cleaning status flow
Non-functional requirements	1. Task list should load within 2 seconds 2. Mobile-friendly interface for on-the-go viewing 3. Clear indication of priority tasks 4. Support for task sorting by location or priority
Postconditions	1. Housekeeper can view assigned tasks 2. Housekeeper can start working on selected task 3. Housekeeper is aware of priorities and special instructions

UC_19: Generate Invoice

UC Name	Generate Invoice
Dependency	US_02 (Make a reservation) or US_03 (Order extra services) or US_08 (Check-Out guest)
Actors	System
Preconditions	1. Reservation exists in system 2. Triggering event occurs (reservation creation, service order, checkout)
Description of Main Sequence	1. System detects invoice generation trigger 2. System retrieves reservation information 3. System calculates room charges 4. System adds service charges 5. System calculates subtotal 6. System applies taxes and fees 7. System calculates total amount 8. System links invoice to reservation 9. System updates invoice status 10. System makes invoice available for viewing
Description of Alternative Sequence	1. Calculation error: a. System logs error b. System alerts administrator 2. Reservation update after invoice: a. System regenerates or updates invoice
Non-functional requirements	1. Invoice generation should complete within 2 seconds 2. Invoice calculations must be accurate to two decimal places 3. System must maintain invoice history 4. Invoice must be formatted according to legal requirements
Postconditions	1. Invoice is created in system 2. Invoice is linked to reservation 3. Invoice is available for payment processing 4. Invoice is available for viewing by staff and guest

UC_20: View available room types

UC Name	View available room types
Dependency	None
Actors	Guest

Preconditions	1. Guest is on booking/reservation page 2. Room types exist in system
Description of Main Sequence	1. Guest enters check-in date 2. Guest enters check-out date 3. Guest specifies number of guests 4. Guest submits search 5. System validates inputs 6. System checks room availability 7. System displays available room types 8. Guest reviews room options
Description of Alternative Sequence	1. Invalid dates: a. System shows date error message b. Guest corrects dates 2. No rooms available: a. System notifies guest b. System suggests alternative dates
Non-functional requirements	1. Availability search should complete within 3 seconds 2. Room information should include high-quality images 3. Clear display of room features and amenities 4. Mobile-friendly responsive design
Postconditions	1. Guest can view available room types 2. Guest can compare room options 3. Guest can proceed to reservation if desired

UC_21: Processing payment

UC Name	Processing payment
Dependency	US_19 (Generate Invoice)
Actors	System, Guest or Receptionist
Preconditions	1. Invoice exists with status "Due" 2. Payment is initiated by guest or receptionist
Description of Main Sequence	1. System retrieves invoice details 2. System displays amount due 3. User selects payment method 4. User provides payment details 5. System connects to payment processor 6. Payment processor authorizes transaction 7. System records payment 8. System updates invoice status 9. System generates receipt

Description of Alternative Sequence	1. Payment declined: a. System displays error message b. User tries alternative payment method 2. Connection error: a. System retries connection b. System suggests alternative payment
Non-functional requirements	1. Payment processing should be secure (PCI compliant) 2. Transaction response within 10 seconds 3. Support for multiple payment methods 4. Clear error messages for failed transactions
Postconditions	1. Payment is processed 2. Invoice status updated to "Paid" 3. Receipt generated for guest 4. Transaction record maintained in system

UC_22: View maintenance request

UC Name	View maintenance request
Dependency	US_01 (User logs in) US_16 (Report an issue) or US_17 (Update cleaning status)
Actors	Manager, Receptionist, Housekeeper
Preconditions	1. Staff member is authenticated 2. Maintenance requests exist in system
Description of Main Sequence	1. Staff member navigates to maintenance section 2. System displays maintenance requests 3. Staff can filter/sort requests as needed 4. Staff selects specific request 5. System displays request details 6. Staff reviews related information
Description of Alternative Sequence	1. No maintenance requests: a. System displays "No requests" message 2. Staff updates request status: a. If authorized, staff can update status b. System saves status change
Non-functional requirements	1. Request list should load within 2 seconds 2. Interface should highlight urgent requests 3. Filtering options for status, location, and age 4. Clear indication of request assignments
Postconditions	1. Staff can view maintenance requests 2. Staff can update request status if authorized 3. Staff can coordinate maintenance activities

UC_23: Complete maintenance request/feedback

UC Name	Complete maintenance request/feedback
Dependency	US_01 (User logs in) US_22 (View maintenance request)
Actors	Manager, Housekeeper
Preconditions	1. Staff member is authenticated 2. Maintenance request exists with status "Pending" or "In Progress" 3. Staff member is assigned to or authorized for the request
Description of Main Sequence	1. Staff selects request to complete 2. Staff reviews request details 3. Staff performs required maintenance 4. Staff returns to system 5. Staff selects "Mark as Completed" 6. System prompts for completion notes 7. Staff enters work performed and follow-up needed 8. Staff uploads photos of completed work (optional) 9. Staff submits completion form 10. System updates request status to "Completed" 11. System notifies relevant parties
Description of Alternative Sequence	1. Partial completion: a. Staff marks as partially complete b. Staff enters remaining work needed c. System keeps request active 2. Parts/assistance needed: a. Staff updates status to "Waiting for parts" b. System maintains request in queue
Non-functional requirements	1. Mobile-friendly interface for on-site completion 2. Photo upload should support common formats 3. Notification to guest should be prompt 4. System should maintain complete maintenance history
Postconditions	1. Maintenance request marked as completed 2. Issue resolution documented in system 3. Guest notified if applicable 4. Room status updated if necessary 5. Maintenance history updated

UC_24: Update reservation

UC Name	Update reservation
Dependency	US_01 (User logs in) US_05 (View reservation)
Actors	Guest, Receptionist
Preconditions	1. User is authenticated 2. Reservation exists 3. Reservation status allows modifications
Description of Main Sequence	1. User selects reservation to modify 2. System displays editable reservation information 3. User updates reservation dates if needed 4. System checks availability for new dates 5. User changes room type if needed 6. System checks availability of new room type 7. User adds additional services if needed 8. User modifies special requests if needed 9. User confirms all changes 10. System validates modifications 11. System updates reservation details 12. System recalculates total price 13. System updates invoice 14. System sends confirmation of changes
Description of Alternative Sequence	1. New dates not available: a. System displays unavailability message b. User selects alternative dates or keeps original 2. New room type not available: a. System displays unavailability message b. User keeps original room type 3. Reservation in non-modifiable state: a. System notifies user that changes are restricted b. User contacts staff for assistance
Non-functional requirements	1. Availability check should complete within 2 seconds 2. Interface must clearly show original and new details 3. System must maintain reservation history 4. Confirmation of changes must be immediate 5. Email notification must include all relevant changes
Postconditions	1. Reservation is updated with changes 2. Invoice is updated if necessary 3. Room availability is updated accordingly 4. Confirmation is sent to guest 5. System logs modification history

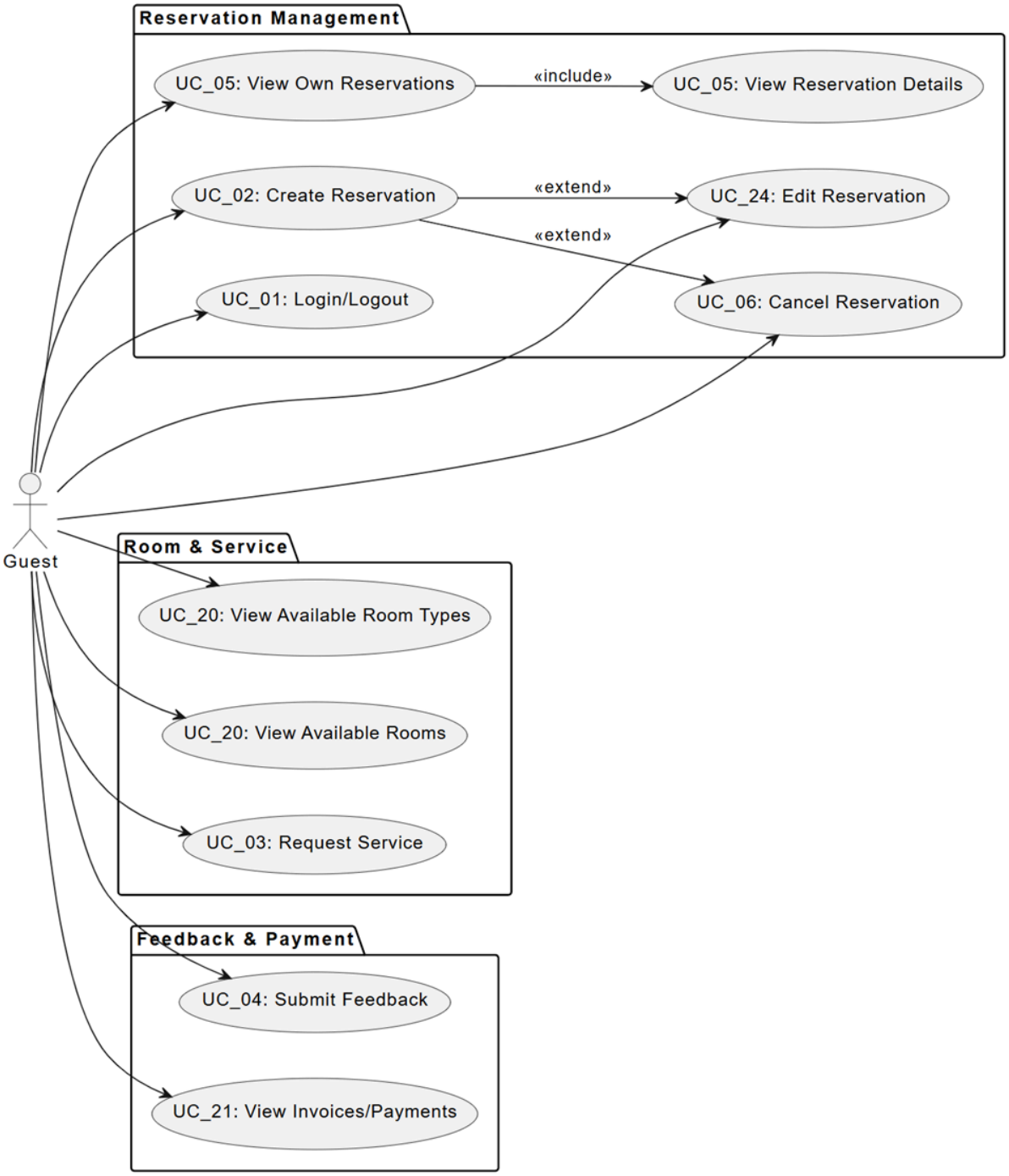
UC_25: Manage rooms

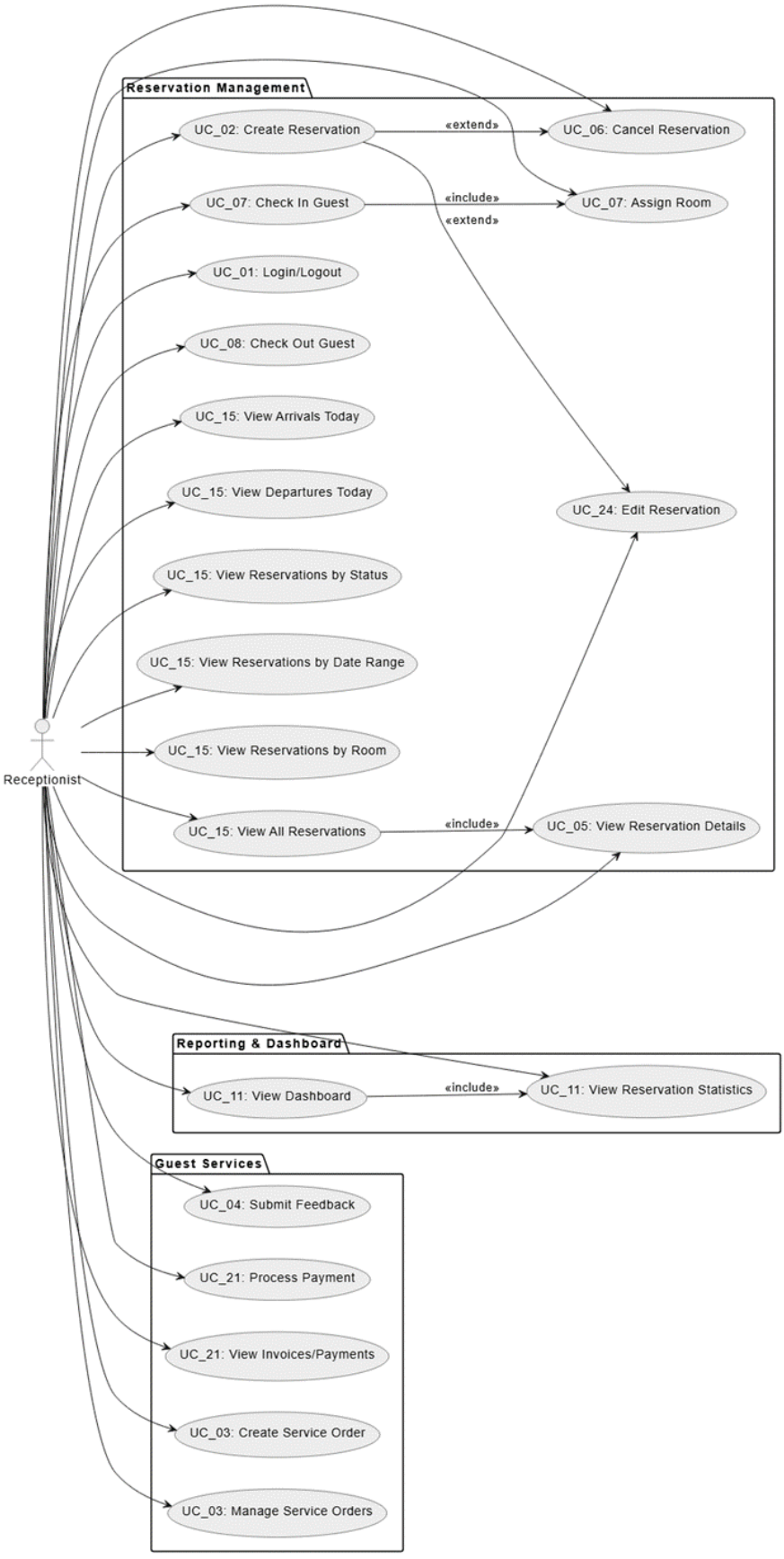
UC Name	Manage rooms
Dependency	US_01 (User logs in)
Actors	Manager
Preconditions	1. Manager is authenticated 2. Manager has room management permissions
Description of Main Sequence	For adding a room: 1. Manager navigates to room management 2. Manager clicks "Add Room" button 3. Manager enters room details (number, type, floor, features) 4. Manager submits form 5. System validates information 6. System creates new room For modifying a room: 1. Manager selects room to modify 2. Manager updates room information 3. Manager submits changes 4. System saves updated details For removing a room: 1. Manager selects room to remove 2. System checks if room has active reservations 3. Manager confirms deletion 4. System removes room from inventory
Description of Alternative Sequence	1. Invalid room information: a. System highlights errors b. Manager corrects information 2. Room has active reservations: a. System prevents deletion b. Manager must relocate guests first 3. Duplicate room number: a. System shows error b. Manager enters different room number
Non-functional requirements	1. Room management interface must be intuitive 2. System must prevent accidental deletion 3. Room status must be clearly indicated 4. System must maintain room history 5. Room images and floor plans should be supported

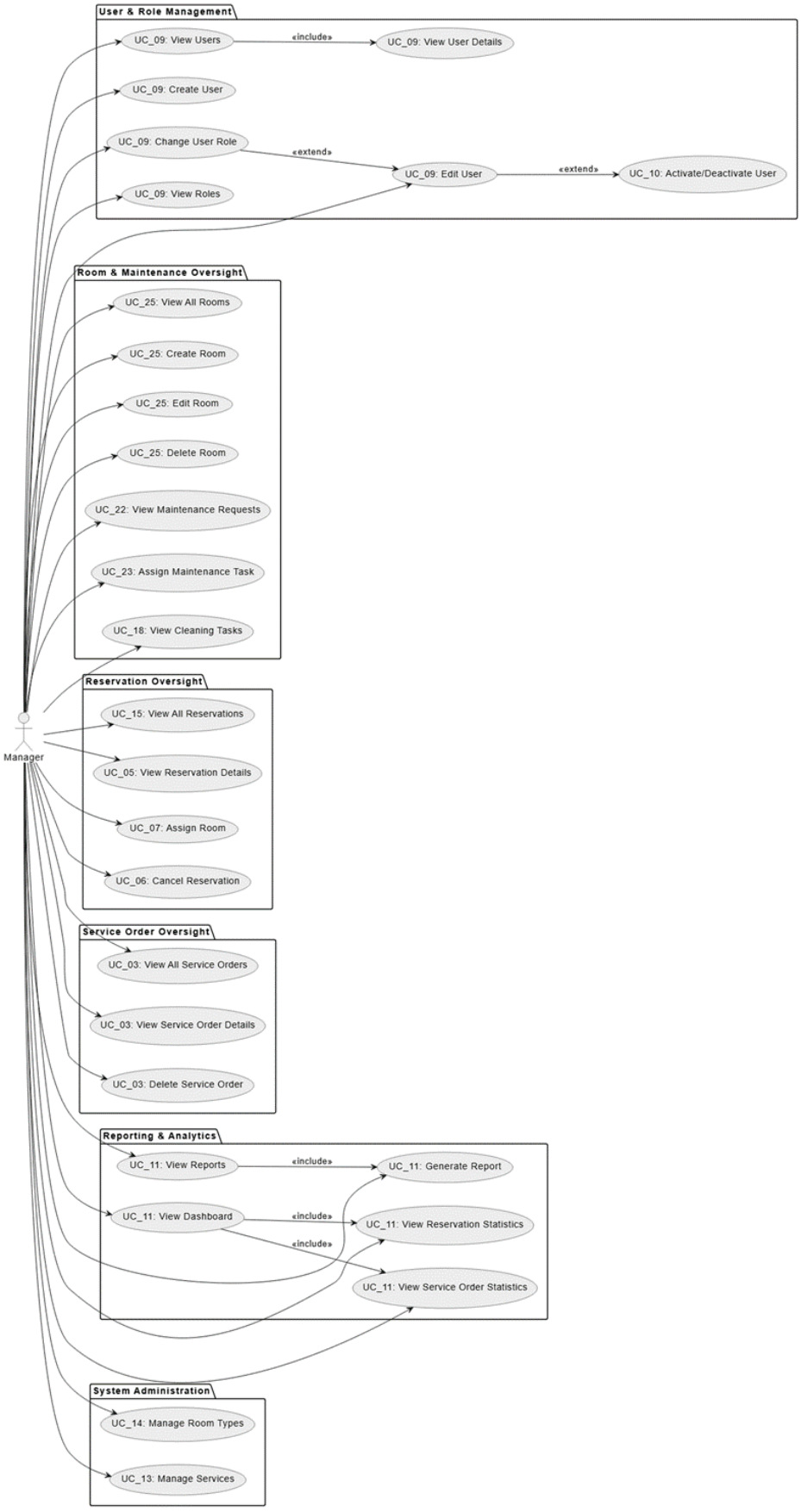
Postconditions	For adding a room: 1. New room is created in system 2. Room is available for booking For modifying a room: 1. Room details are updated 2. Changes are reflected in booking system For removing a room: 1. Room is removed from inventory 2. Room is no longer available for booking

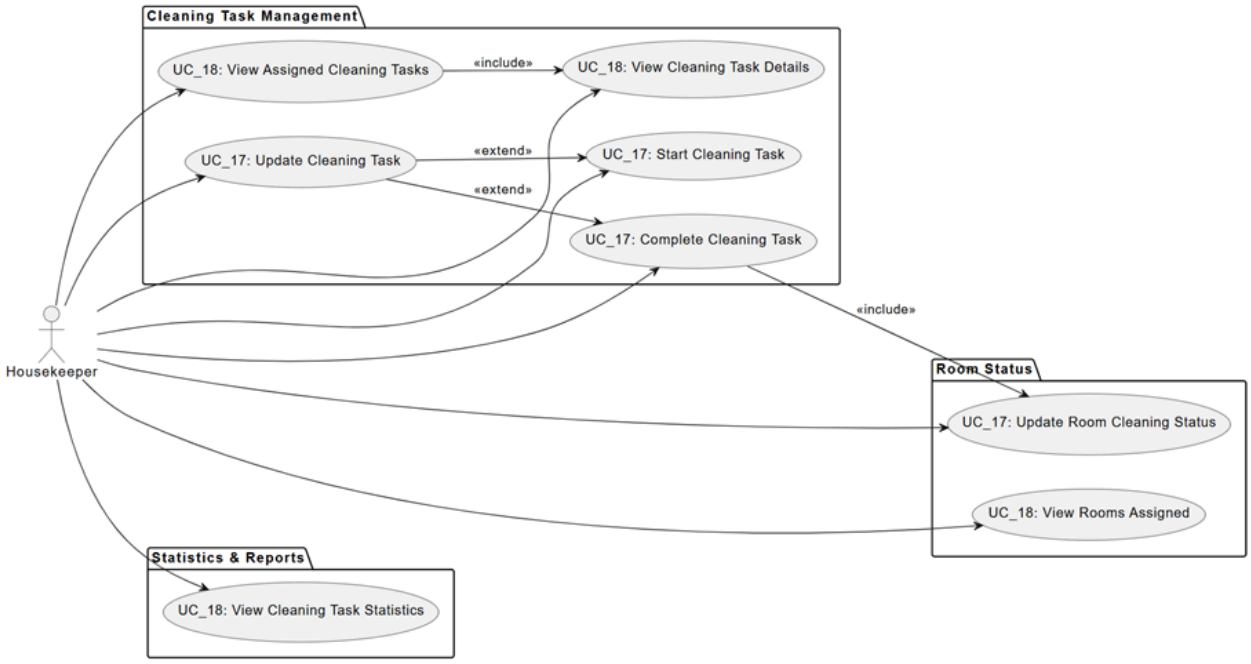
4.2 Behavioral Diagrams

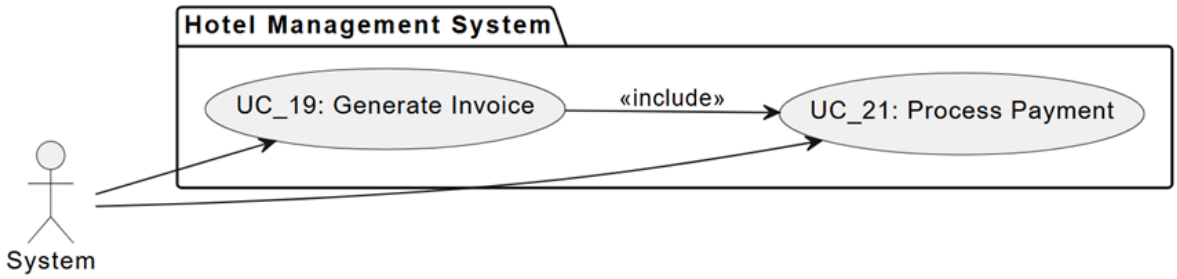
4.2.1 Use Case Diagrams


Reservation Management
UC_05: View Own Reservations
«include»
UC_05: View Reservation Details
UC_02: Create Reservation
«extend»
UC_24: Edit Reservation
«extend»
UC_01: Login/Logout
UC_06: Cancel Reservation
Guest
Room & Service
UC_20: View Available Room Types
UC_20: View Available Rooms
UC_03: Request Service
Feedback & Payment
UC_04: Submit Feedback
UC_21: View Invoices/Payments

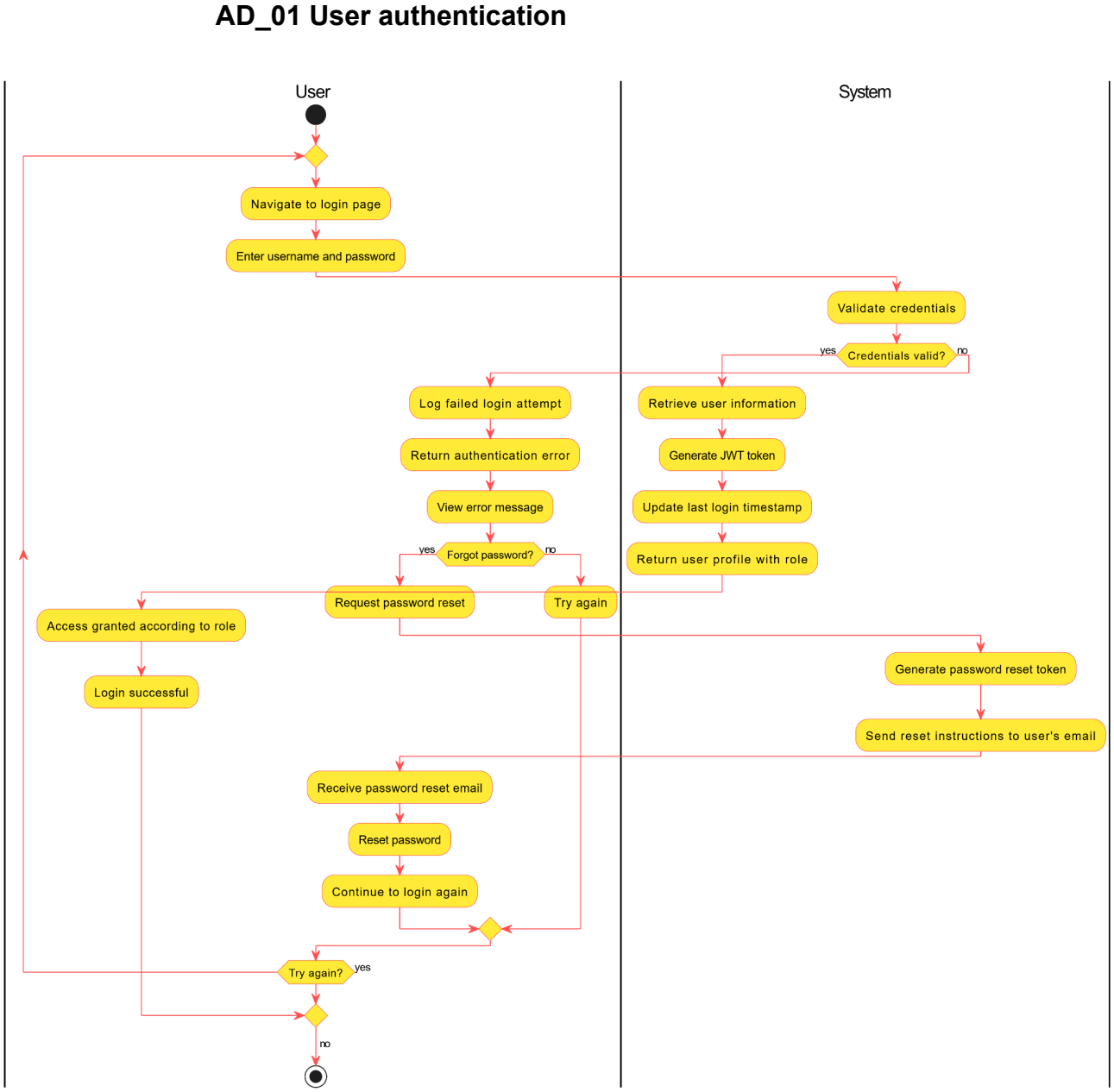

Reservation Management
UC_02: Create Reservation
«extend»
UC_06: Cancel Reservation
UC_07: Check In Guest
«include»
«extend»
UC_07: Assign Room
UC_01: Login/Logout
UC_08: Check Out Guest
UC_15: View Arrivals Today
UC_15: View Departures Today
UC_24: Edit Reservation
UC_15: View Reservations by Status
UC_15: View Reservations by Date Range
UC_15: View Reservations by Room
UC_15: View All Reservations
«include»
UC_05: View Reservation Details
Receptionist
Reporting & Dashboard
UC_11: View Dashboard
«include»
UC_11: View Reservation Statistics
Guest Services
UC_04: Submit Feedback
UC_21: Process Payment
UC_21: View Invoices/Payments
UC_03: Create Service Order
UC_03: Manage Service Orders


User & Role Management
UC_09: View Users
«include»
UC_09: View User Details
UC_09: Create User
UC_09: Change User Role
«extend»
UC_09: Edit User
«extend»
UC_10: Activate/Deactivate User
UC_09: View Roles
Room & Maintenance Oversight
UC_25: View All Rooms
UC_25: Create Room
UC_25: Edit Room
UC_25: Delete Room
UC_22: View Maintenance Requests
UC_23: Assign Maintenance Task
UC_18: View Cleaning Tasks
Reservation Oversight
UC_15: View All Reservations
UC_05: View Reservation Details
UC_07: Assign Room
UC_06: Cancel Reservation
Service Order Oversight
UC_03: View All Service Orders
UC_03: View Service Order Details
UC_03: Delete Service Order
Reporting & Analytics
UC_11: View Reports
«include»
UC_11: Generate Report
UC_11: View Dashboard
«include»
UC_11: View Reservation Statistics
«include»
UC_11: View Service Order Statistics
System Administration
UC_14: Manage Room Types
UC_13: Manage Services
Manager

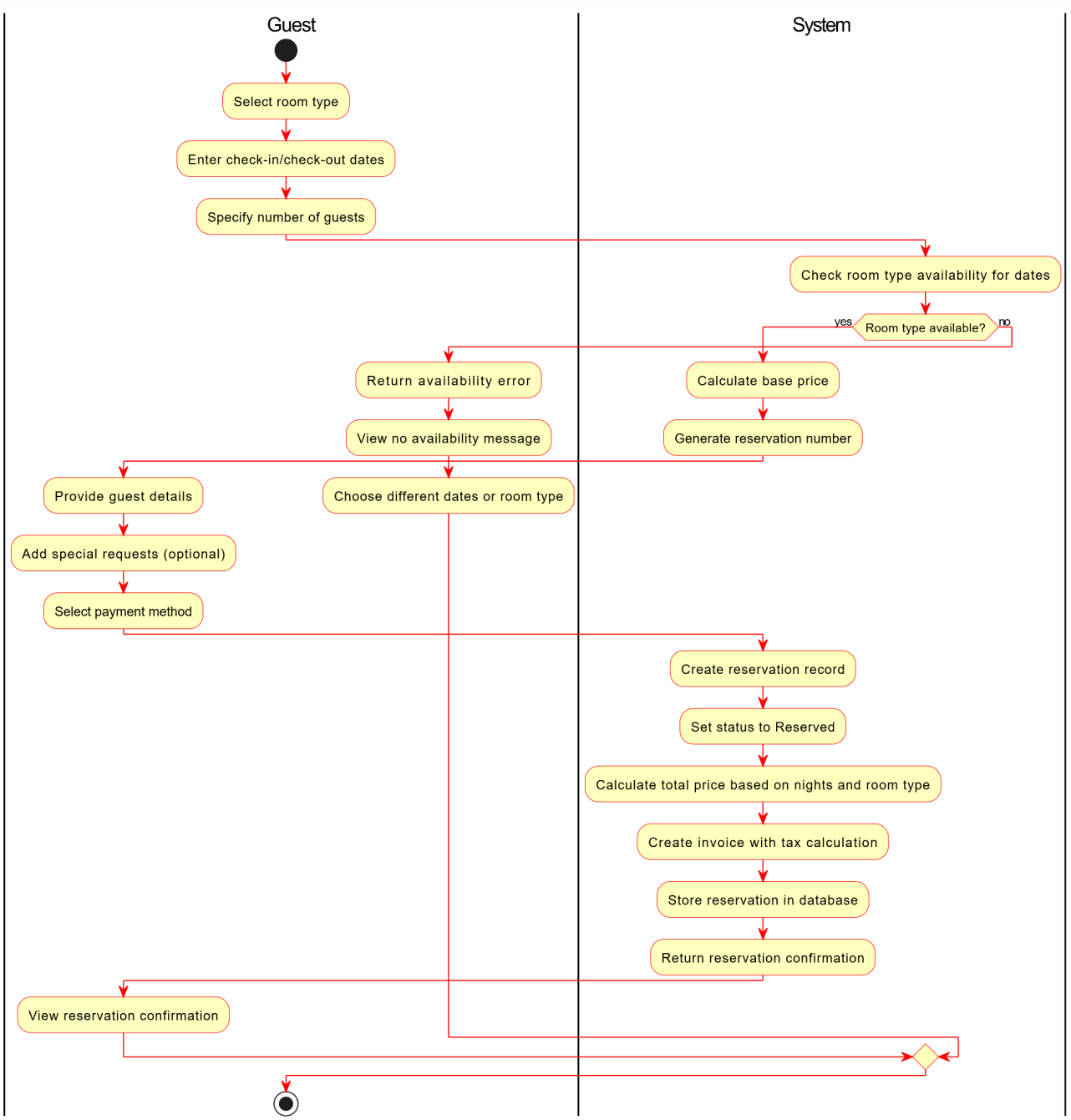

Cleaning Task Management
UC_18: View Assigned Cleaning Tasks
«include»
UC_18: View Cleaning Task Details
UC_17: Update Cleaning Task
«extend»
UC_17: Start Cleaning Task
«extend»
UC_17: Complete Cleaning Task
«include»
Housekeeper
Room Status
UC_17: Update Room Cleaning Status
UC_18: View Rooms Assigned
Statistics & Reports
UC_18: View Cleaning Task Statistics

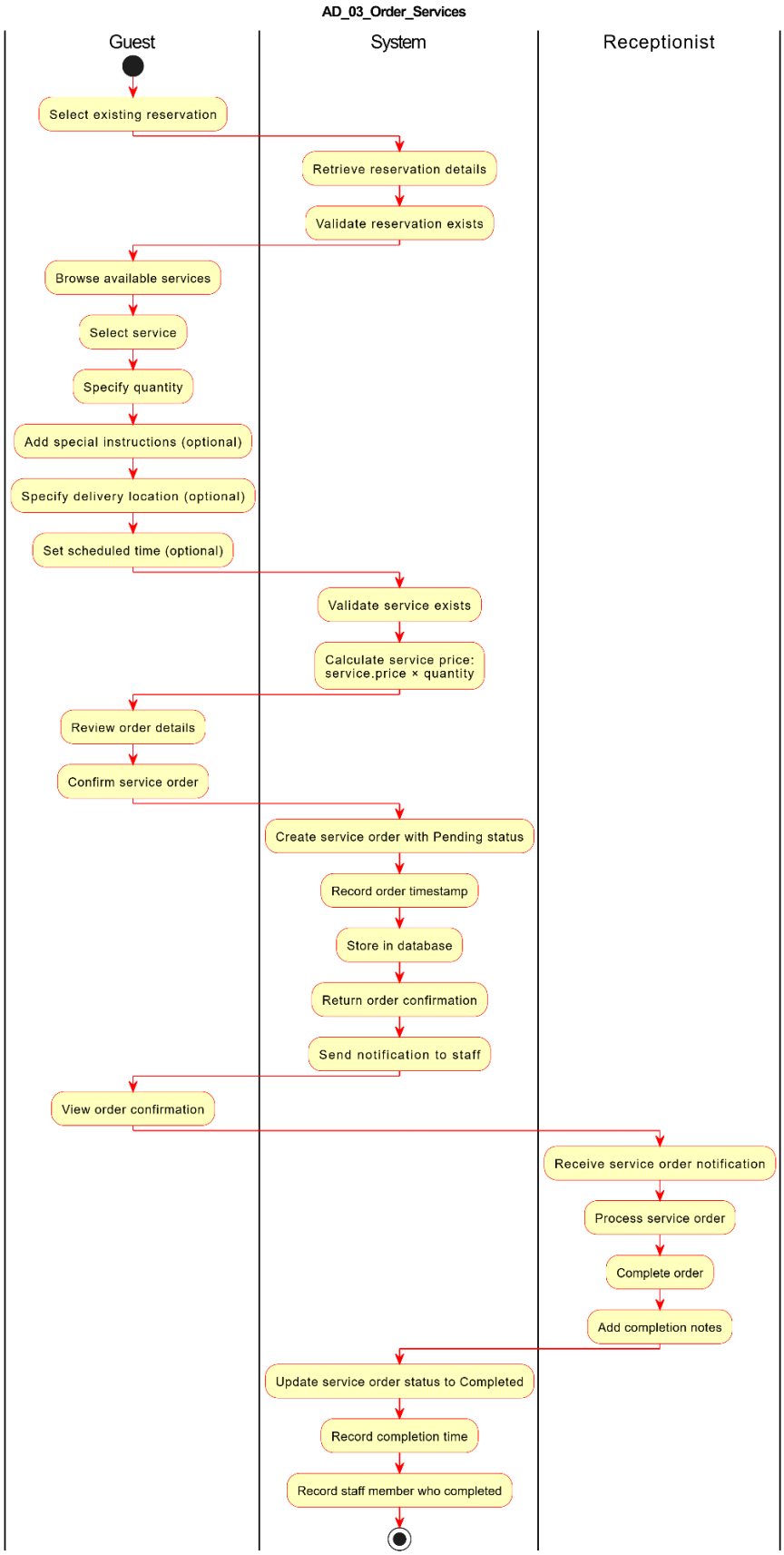

Hotel Management System
UC_19: Generate Invoice
«include»
UC_21: Process Payment
System

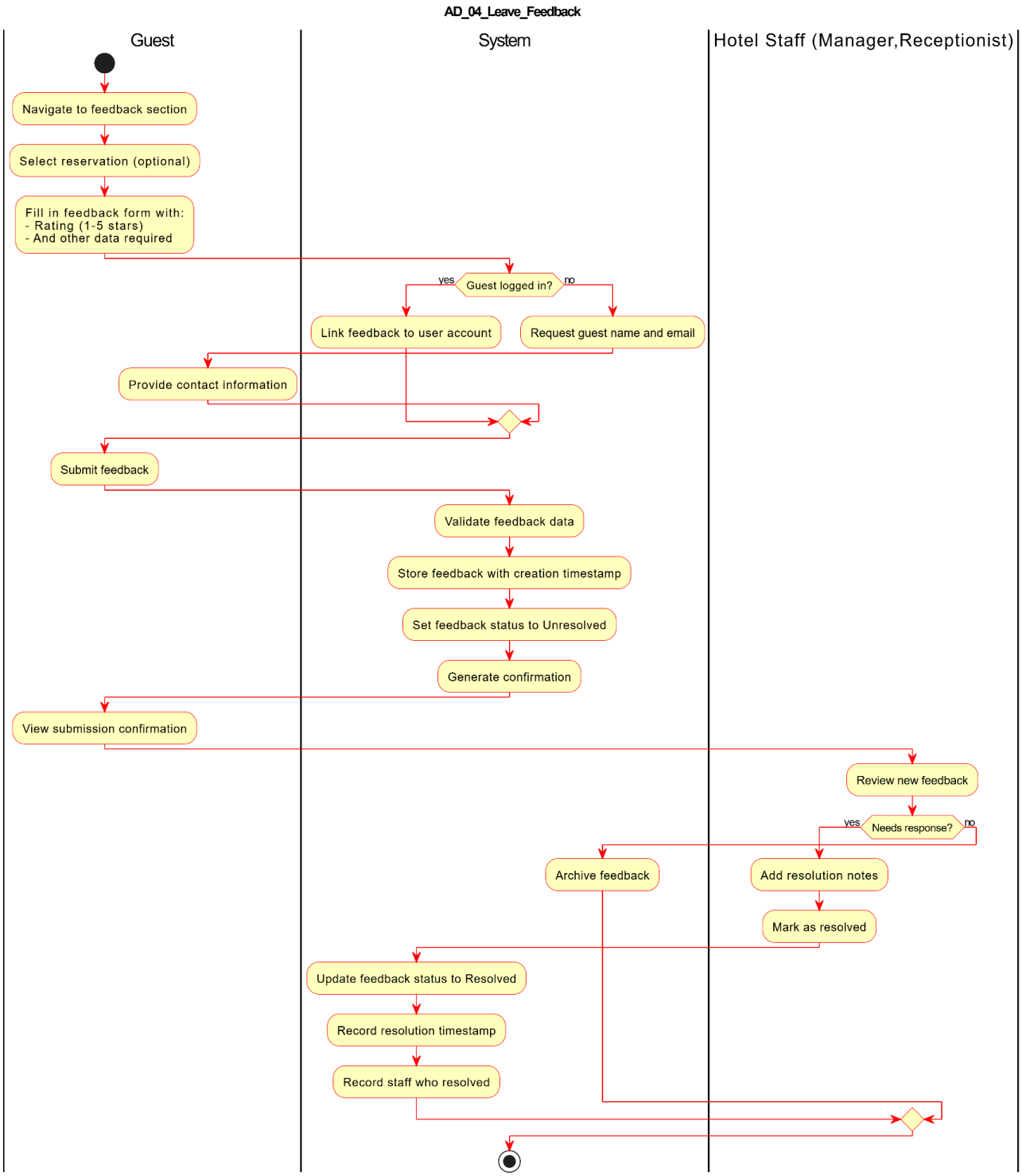
4.2.2 Activity Diagrams

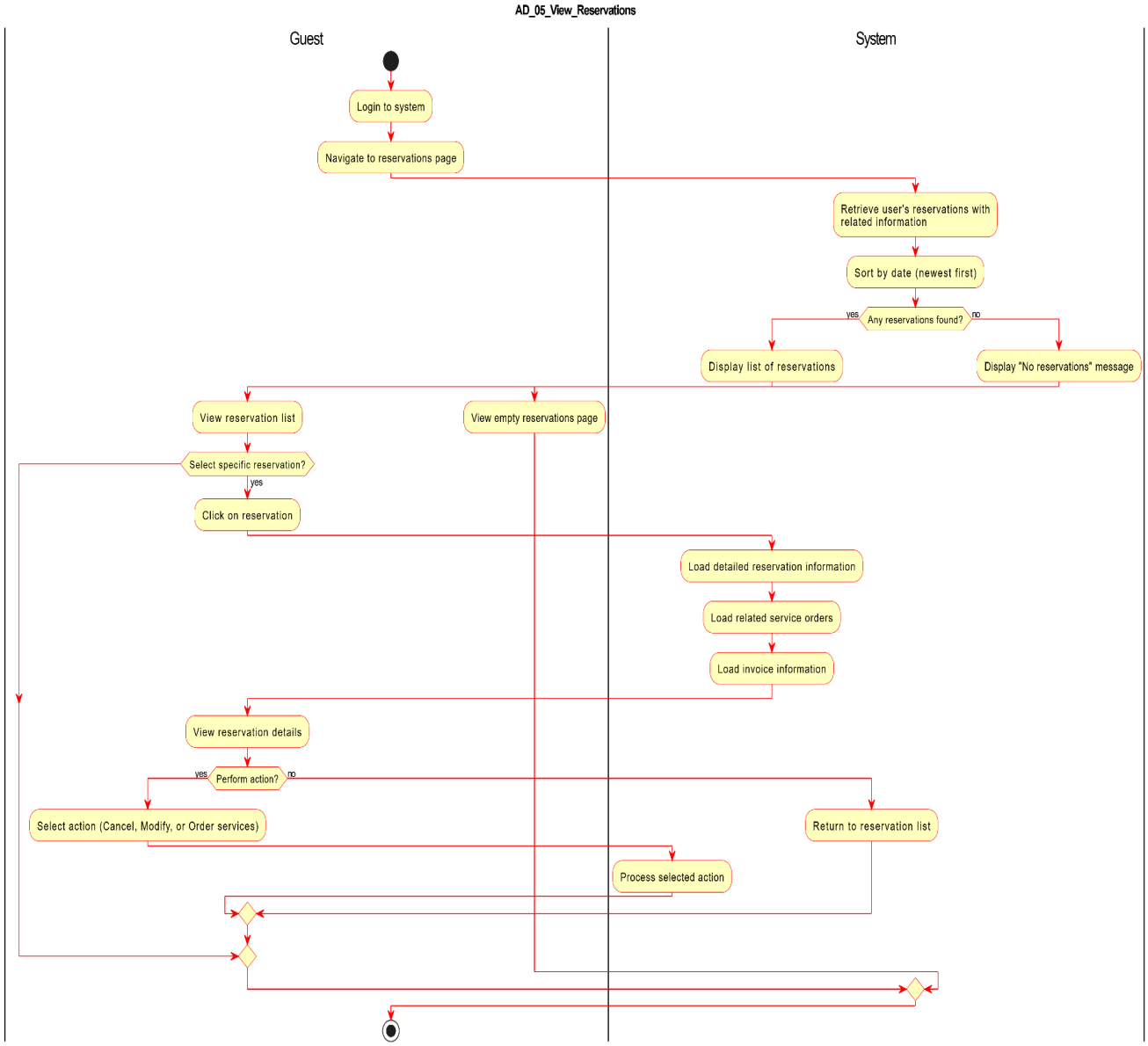


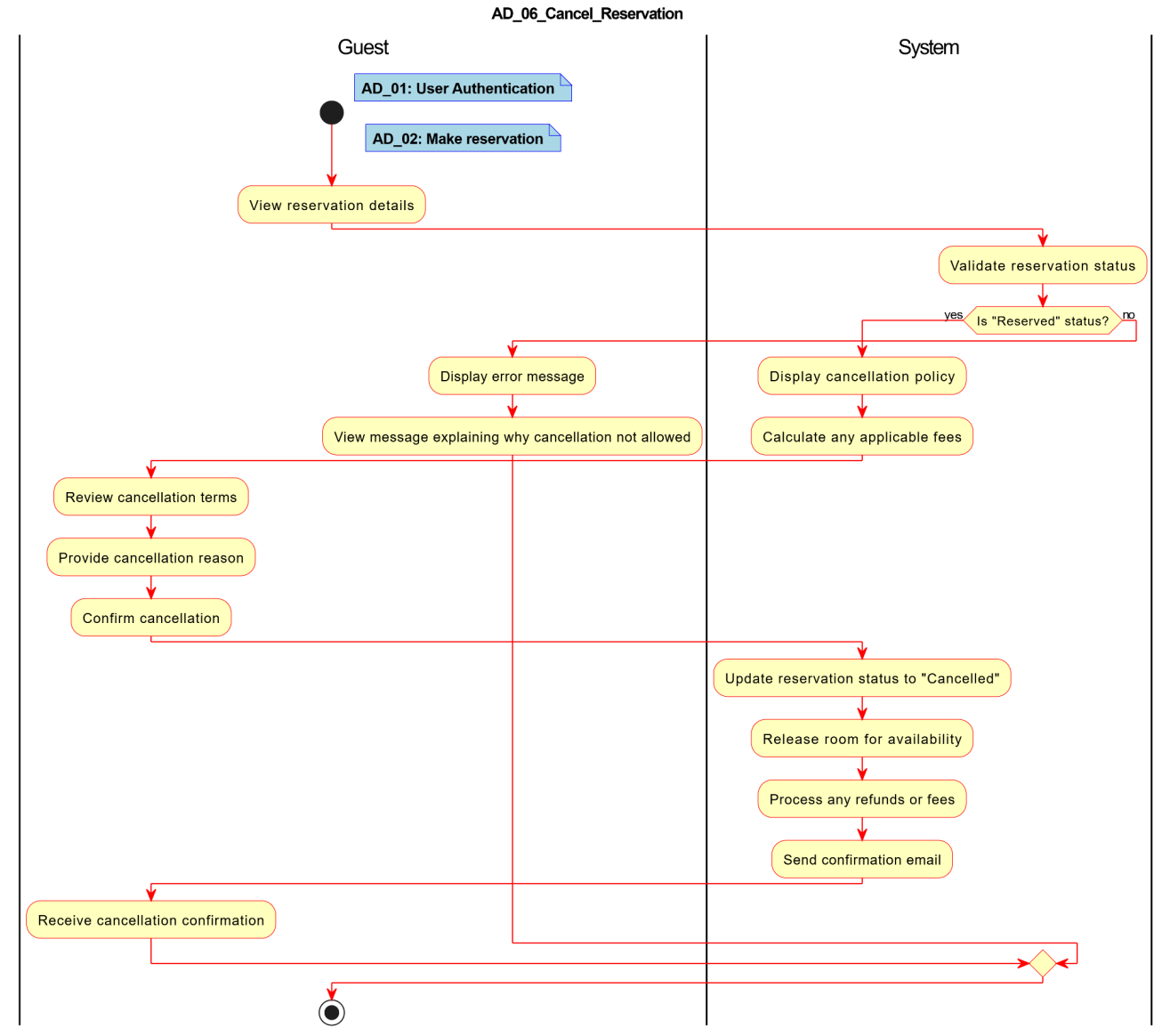
AD_02 Making a reservation

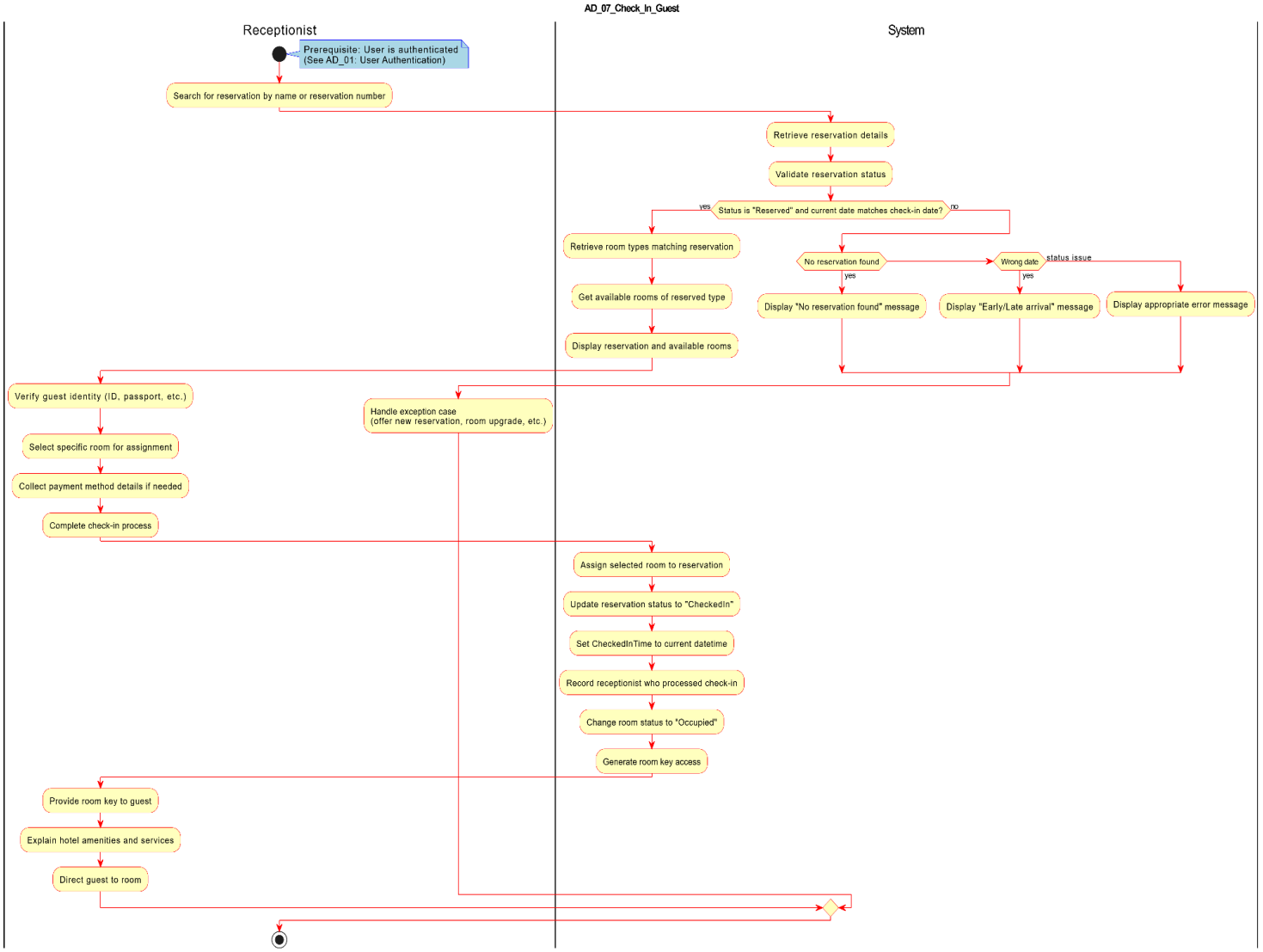


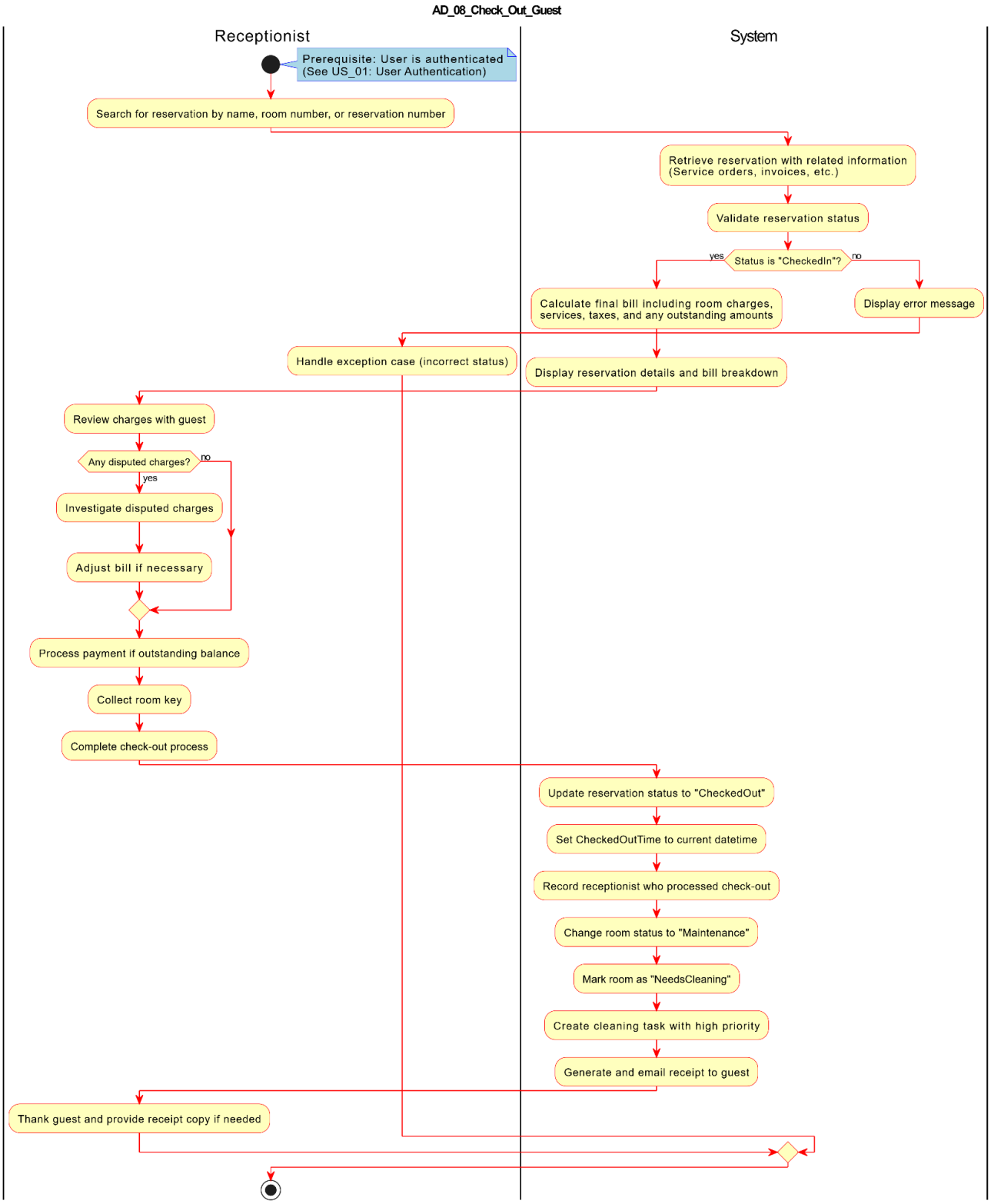

AD_03_Order_Services
Guest
System
Receptionist
Select existing reservation
Retrieve reservation details
Validate reservation exists
Browse available services
Select service
Specify quantity
Add special instructions (optional)
Specify delivery location (optional)
Set scheduled time (optional)
Validate service exists
Calculate service price: service.price × quantity
Review order details
Confirm service order
Create service order with Pending status
Record order timestamp
Store in database
Return order confirmation
Send notification to staff
View order confirmation
Receive service order notification
Process service order
Complete order
Add completion notes
Update service order status to Completed
Record completion time
Record staff member who completed

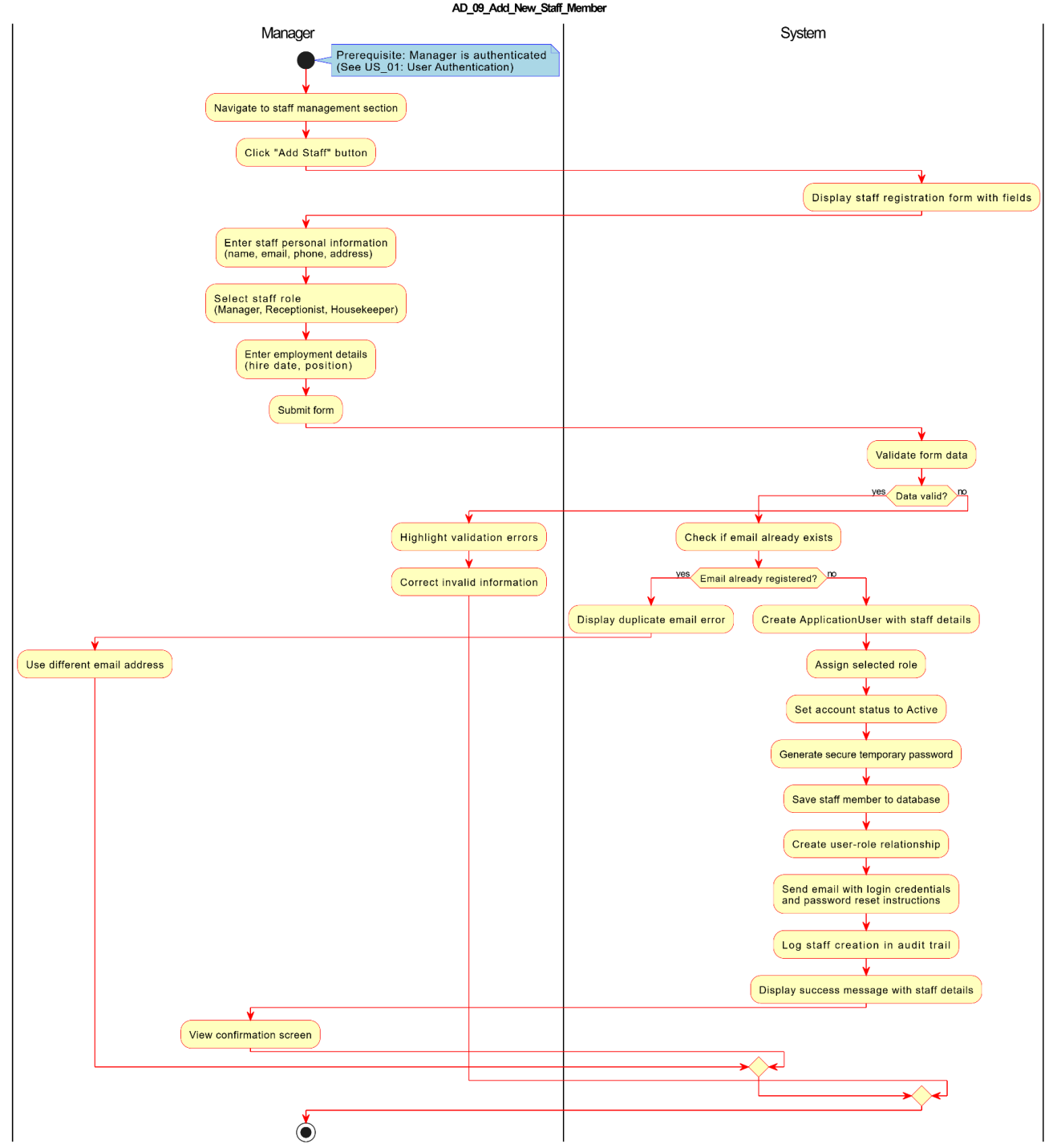

AD_04_Leave_Feedback
Guest
System
Hotel Staff (Manager,Receptionist)
Navigate to feedback section
Select reservation (optional)
Fill in feedback form with:
- Rating (1-5 stars)
- And other data required
yes
Guest logged in?
no
Link feedback to user account
Request guest name and email
Provide contact information
Submit feedback
Validate feedback data
Store feedback with creation timestamp
Set feedback status to Unresolved
Generate confirmation
View submission confirmation
Review new feedback
yes
Needs response?
no
Archive feedback
Add resolution notes
Mark as resolved
Update feedback status to Resolved
Record resolution timestamp
Record staff who resolved

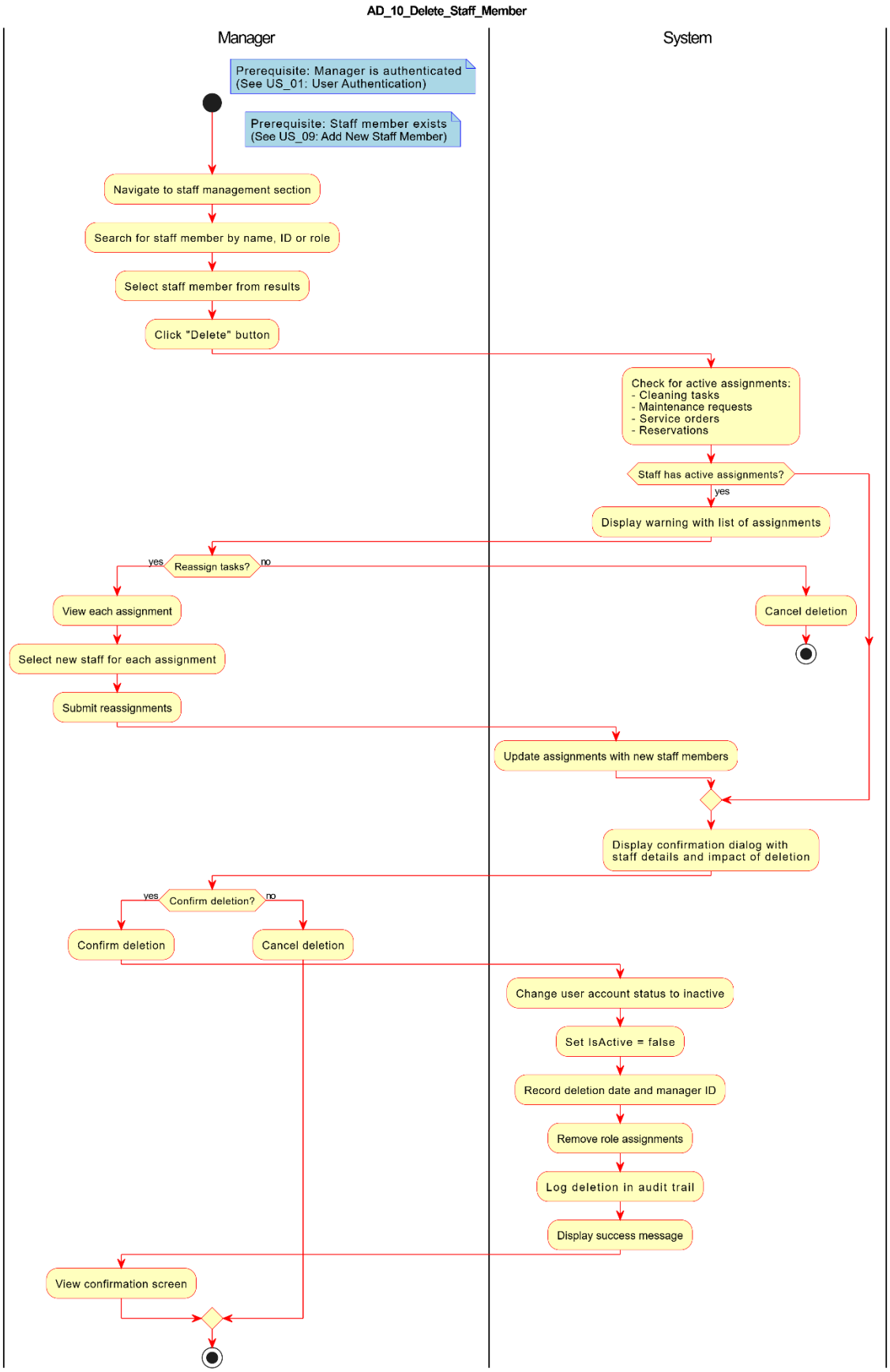

AD_05_View_Reservations
Guest
System
Login to system
Navigate to reservations page
Retrieve user's reservations with related information
Sort by date (newest first)
Any reservations found?
yes
no
Display list of reservations
Display "No reservations" message
View reservation list
View empty reservations page
Select specific reservation?
yes
Click on reservation
Load detailed reservation information
Load related service orders
Load invoice information
View reservation details
Perform action?
yes
no
Select action (Cancel, Modify, or Order services)
Return to reservation list
Process selected action

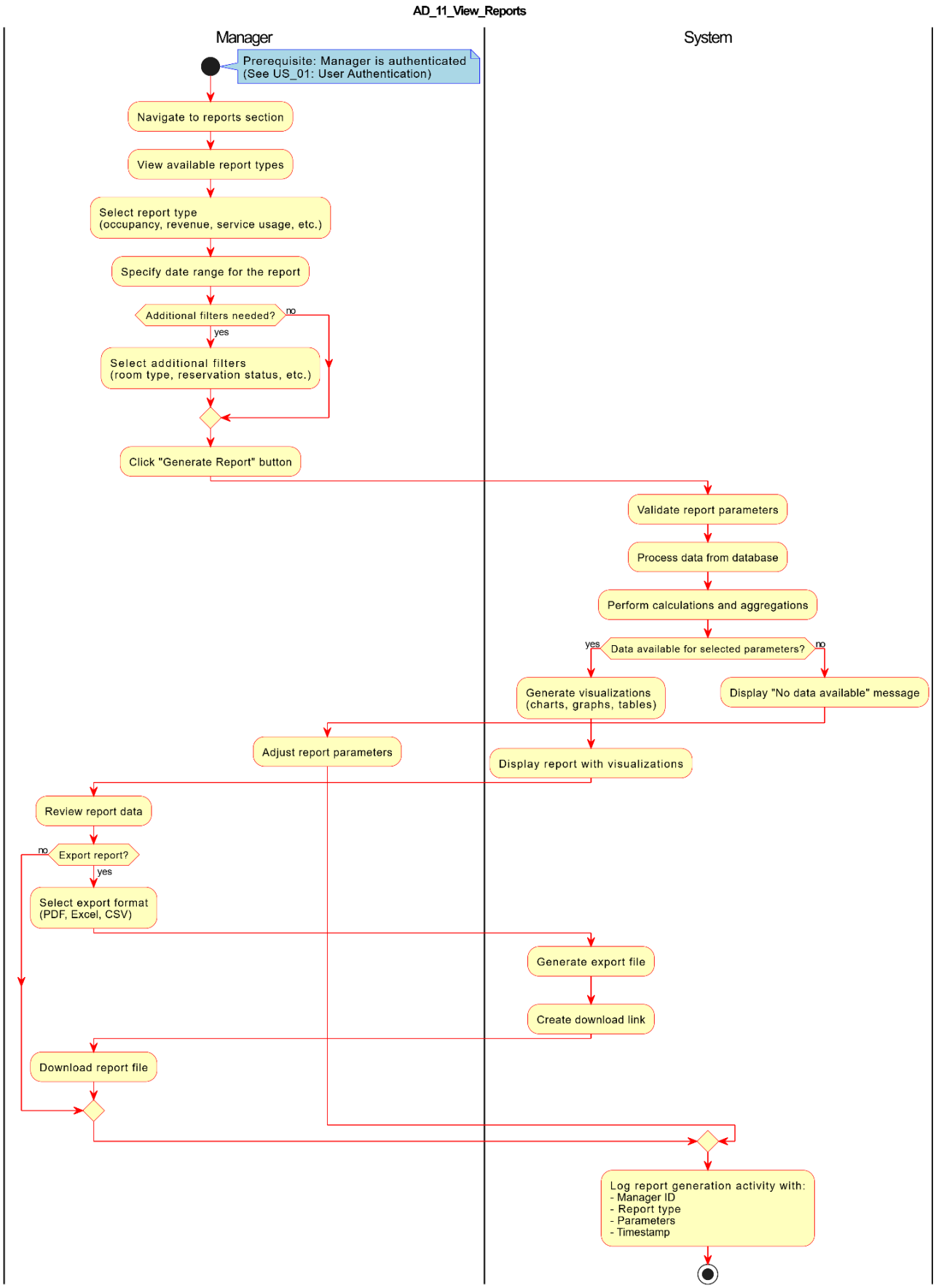

AD_06_Cancel_Reservation
Guest
System
AD_01: User Authentication
AD_02: Make reservation
View reservation details
Validate reservation status
Is "Reserved" status?
yes
no
Display error message
View message explaining why cancellation not allowed
Display cancellation policy
Calculate any applicable fees
Review cancellation terms
Provide cancellation reason
Confirm cancellation
Update reservation status to "Cancelled"
Release room for availability
Process any refunds or fees
Send confirmation email
Receive cancellation confirmation

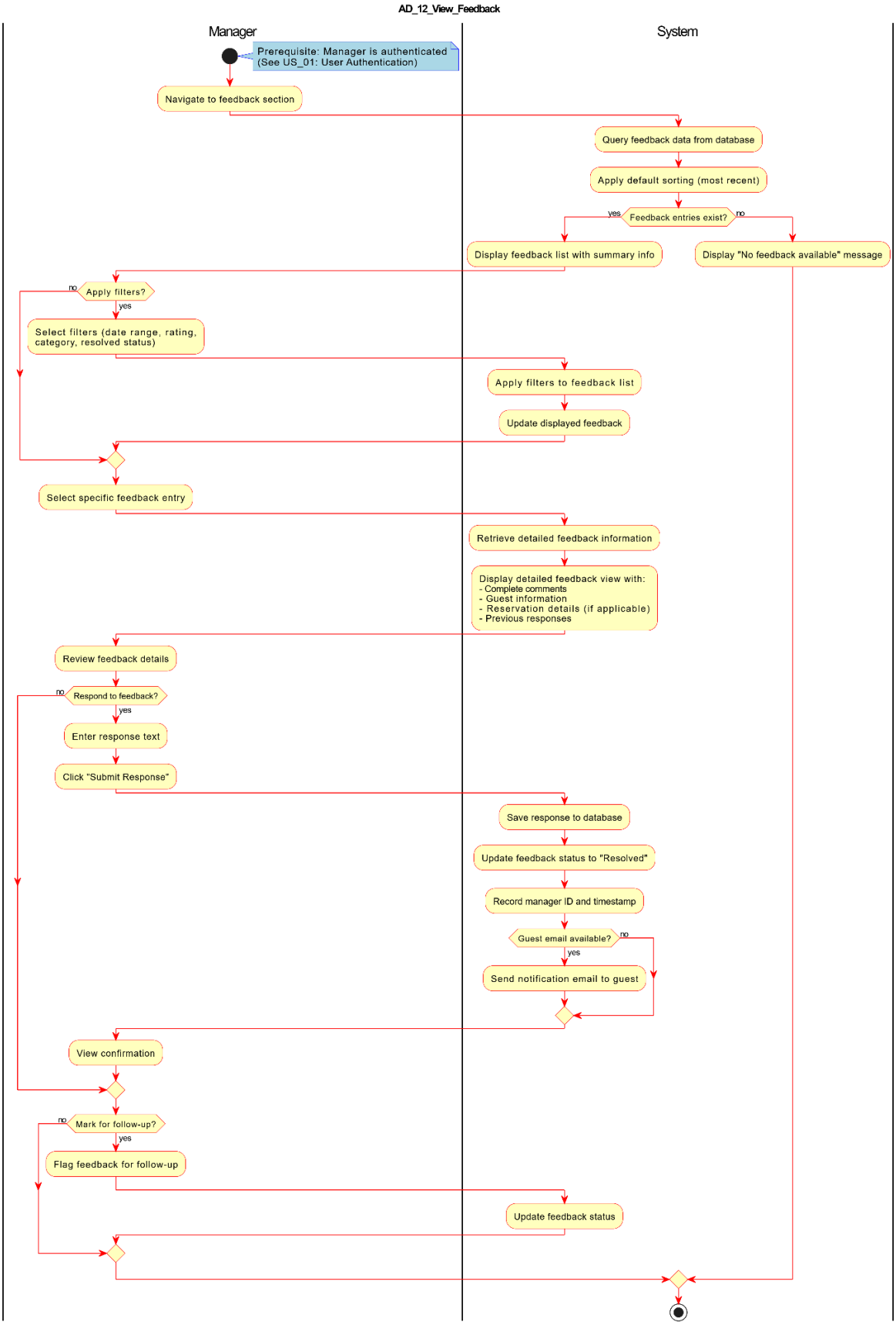

AD_07_Check_In_Guest
Receptionist
System
Prerequisite: User is authenticated (See AD_01: User Authentication)
Search for reservation by name or reservation number
Retrieve reservation details
Validate reservation status
yes
Status is "Reserved" and current date matches check-in date?
no
Retrieve room types matching reservation
No reservation found
yes
Wrong date
status issue
yes
Get available rooms of reserved type
Display "No reservation found" message
Display "Early/Late arrival" message
Display appropriate error message
Display reservation and available rooms
Verify guest identity (ID, passport, etc.)
Handle exception case (offer new reservation, room upgrade, etc.)
Select specific room for assignment
Collect payment method details if needed
Complete check-in process
Assign selected room to reservation
Update reservation status to "CheckedIn"
Set CheckedInTime to current datetime
Record receptionist who processed check-in
Change room status to "Occupied"
Generate room key access
Provide room key to guest
Explain hotel amenities and services
Direct guest to room

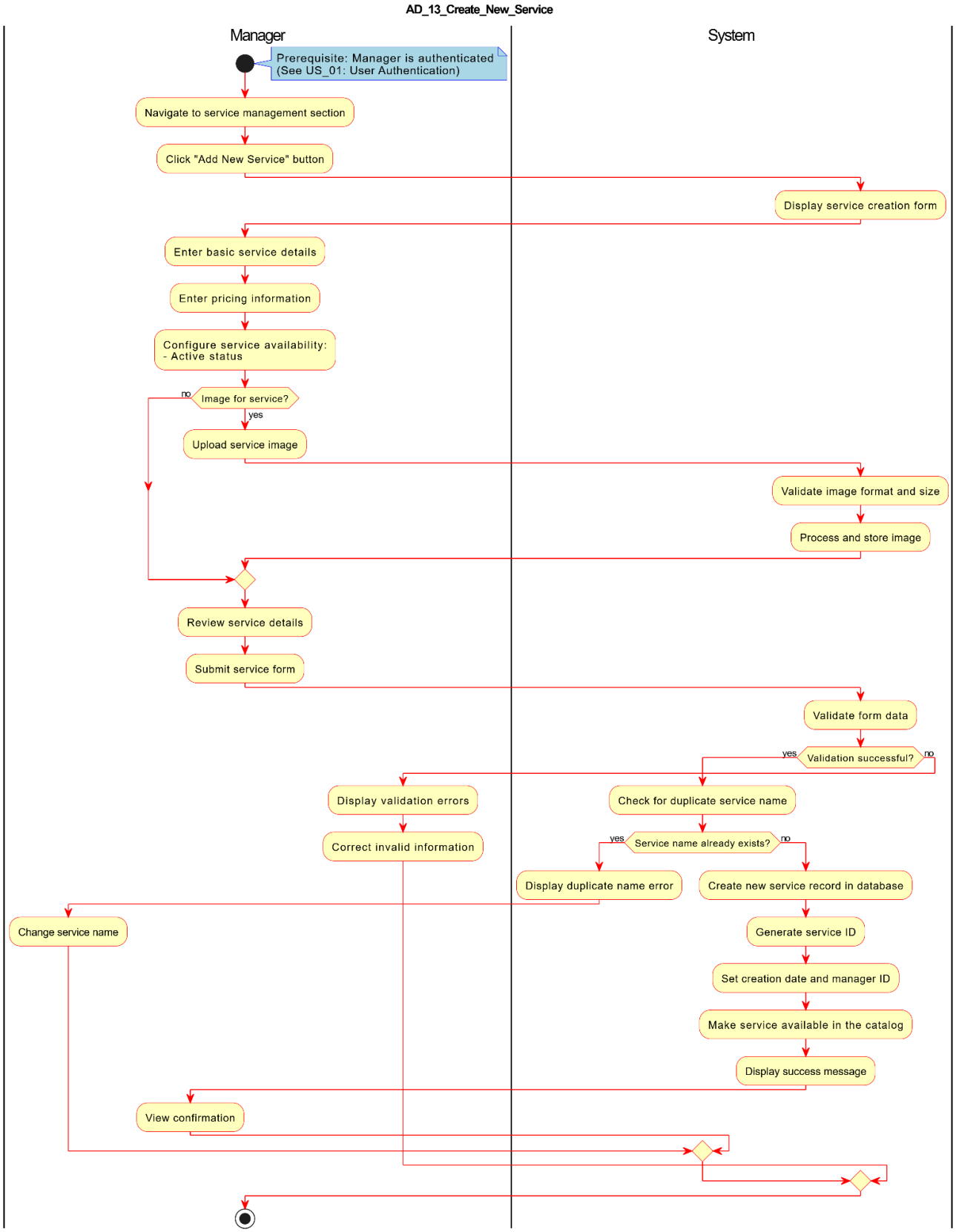

AD_08_Check_Out_Guest
Receptionist
System
Prerequisite: User is authenticated (See US_01: User Authentication)
Search for reservation by name, room number, or reservation number
Retrieve reservation with related information (Service orders, invoices, etc.)
Validate reservation status
Status is "CheckedIn"?
yes
no
Calculate final bill including room charges, services, taxes, and any outstanding amounts
Display error message
Handle exception case (incorrect status)
Display reservation details and bill breakdown
Review charges with guest
Any disputed charges?
no
yes
Investigate disputed charges
Adjust bill if necessary
Process payment if outstanding balance
Collect room key
Complete check-out process
Update reservation status to "CheckedOut"
Set CheckedOutTime to current datetime
Record receptionist who processed check-out
Change room status to "Maintenance"
Mark room as "NeedsCleaning"
Create cleaning task with high priority
Generate and email receipt to guest
Thank guest and provide receipt copy if needed

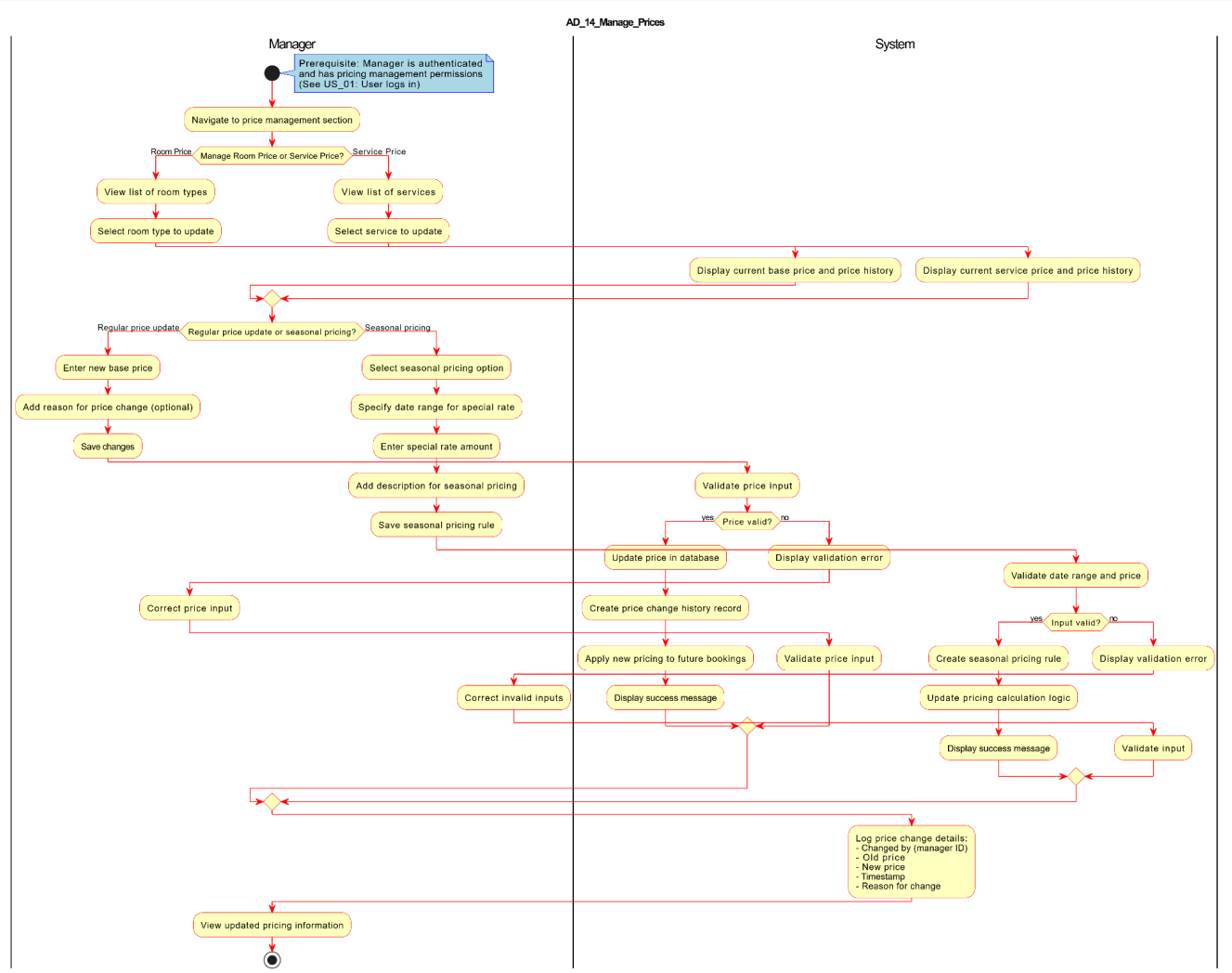

AD_09_Add_New_Staff_Member
Manager
System
Prerequisite: Manager is authenticated (See US_01: User Authentication)
Navigate to staff management section
Click "Add Staff" button
Display staff registration form with fields
Enter staff personal information (name, email, phone, address)
Select staff role (Manager, Receptionist, Housekeeper)
Enter employment details (hire date, position)
Submit form
Validate form data
Data valid?
yes
no
Highlight validation errors
Correct invalid information
Check if email already exists
Email already registered?
yes
no
Display duplicate email error
Create ApplicationUser with staff details
Use different email address
Assign selected role
Set account status to Active
Generate secure temporary password
Save staff member to database
Create user-role relationship
Send email with login credentials and password reset instructions
Log staff creation in audit trail
Display success message with staff details
View confirmation screen

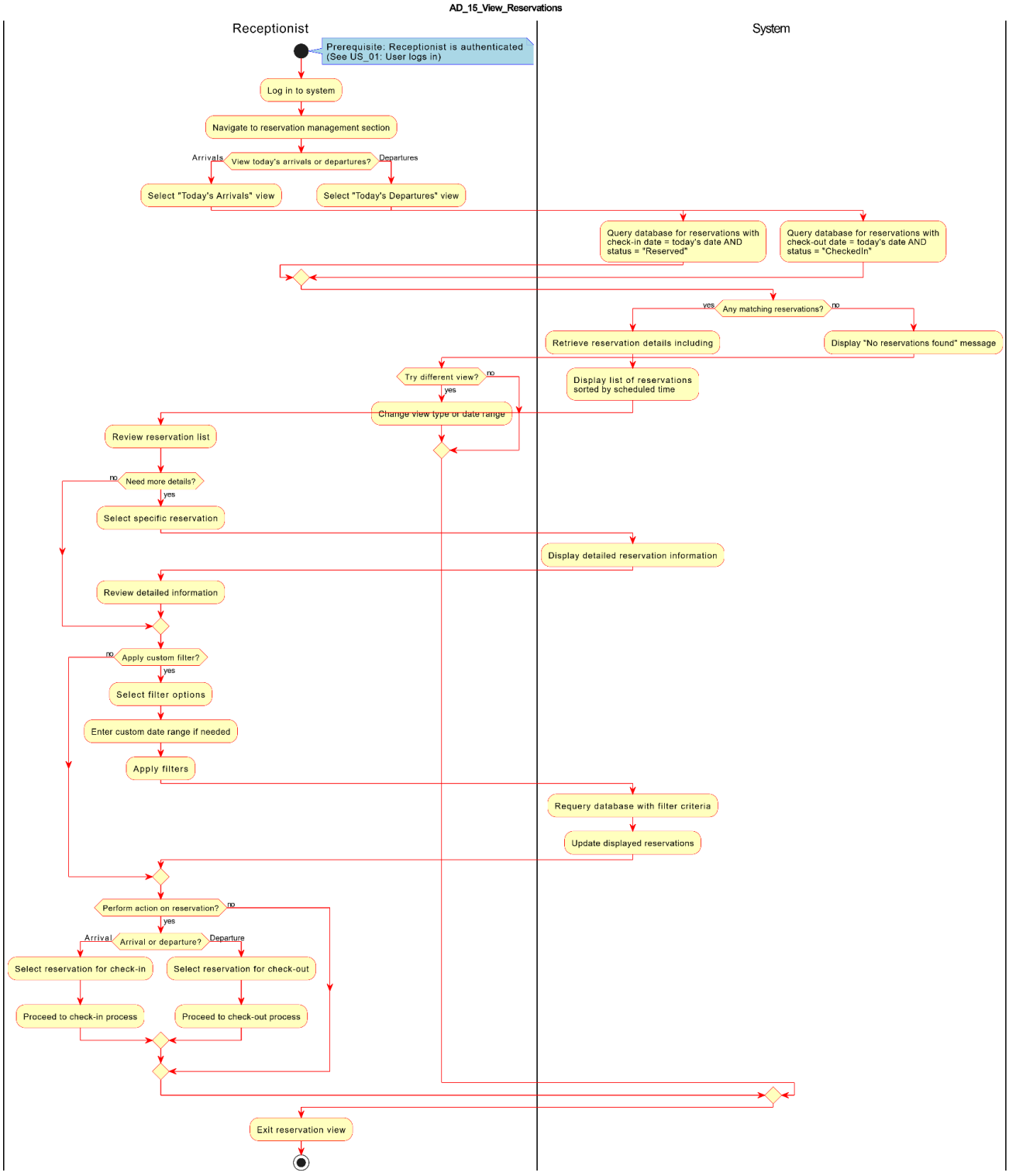

AD_10_Delete_Staff_Member
Manager
System
Prerequisite: Manager is authenticated (See US_01: User Authentication)
Prerequisite: Staff member exists (See US_09: Add New Staff Member)
Navigate to staff management section
Search for staff member by name, ID or role
Select staff member from results
Click "Delete" button
Check for active assignments:
- Cleaning tasks
- Maintenance requests
- Service orders
- Reservations
Staff has active assignments?
yes
Display warning with list of assignments
yes
Reassign tasks?
no
View each assignment
Select new staff for each assignment
Submit reassignments
Cancel deletion
Update assignments with new staff members
Display confirmation dialog with staff details and impact of deletion
yes
Confirm deletion?
no
Confirm deletion
Cancel deletion
Change user account status to inactive
Set IsActive = false
Record deletion date and manager ID
Remove role assignments
Log deletion in audit trail
Display success message
View confirmation screen

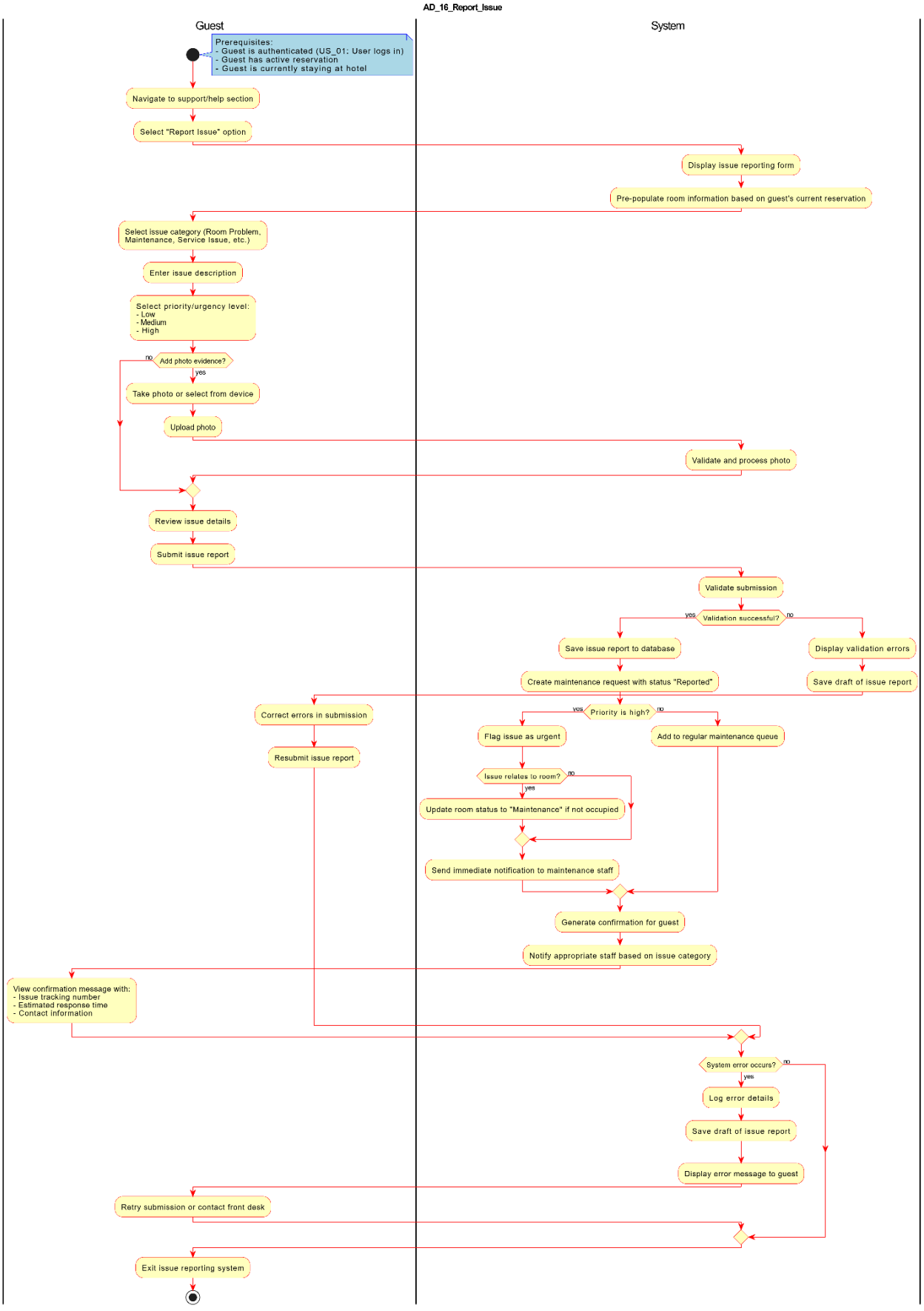

AD_11_View_Reports
Manager
System
Prerequisite: Manager is authenticated
(See US_01: User Authentication)
Navigate to reports section
View available report types
Select report type
(occupancy, revenue, service usage, etc.)
Specify date range for the report
Additional filters needed?
no
yes
Select additional filters
(room type, reservation status, etc.)
Click "Generate Report" button
Validate report parameters
Process data from database
Perform calculations and aggregations
yes
Data available for selected parameters?
no
Generate visualizations
(charts, graphs, tables)
Display "No data available" message
Adjust report parameters
Display report with visualizations
Review report data
no
Export report?
yes
Select export format
(PDF, Excel, CSV)
Generate export file
Create download link
Download report file
Log report generation activity with:
- Manager ID
- Report type
- Parameters
- Timestamp

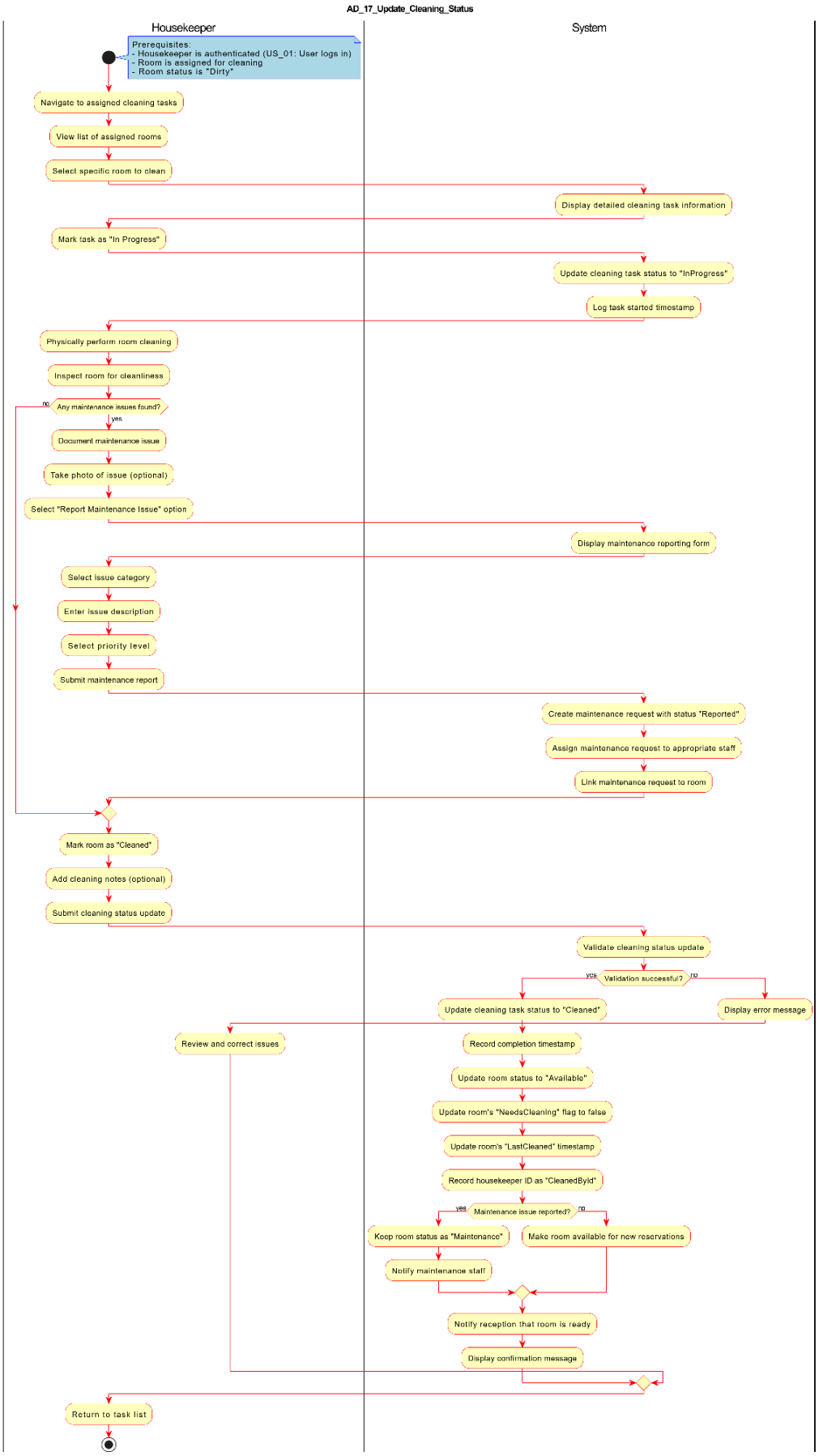

AD_12_View_Feedback
Manager
System
Prerequisite: Manager is authenticated (See US_01: User Authentication)
Navigate to feedback section
Query feedback data from database
Apply default sorting (most recent)
Feedback entries exist?
yes
no
Display feedback list with summary info
Display "No feedback available" message
Apply filters?
no
yes
Select filters (date range, rating, category, resolved status)
Apply filters to feedback list
Update displayed feedback
Select specific feedback entry
Retrieve detailed feedback information
Display detailed feedback view with:
- Complete comments
- Guest information
- Reservation details (if applicable)
- Previous responses
Review feedback details
Respond to feedback?
no
yes
Enter response text
Click "Submit Response"
Save response to database
Update feedback status to "Resolved"
Record manager ID and timestamp
Guest email available?
no
yes
Send notification email to guest
View confirmation
Mark for follow-up?
no
yes
Flag feedback for follow-up
Update feedback status


AD_13_Create_New_Service
Manager
System
Prerequisite: Manager is authenticated (See US_01: User Authentication)
Navigate to service management section
Click "Add New Service" button
Display service creation form
Enter basic service details
Enter pricing information
Configure service availability: - Active status
Image for service?
no
yes
Upload service image
Validate image format and size
Process and store image
Review service details
Submit service form
Validate form data
Validation successful?
yes
no
Display validation errors
Correct invalid information
Check for duplicate service name
Service name already exists?
yes
no
Display duplicate name error
Create new service record in database
Change service name
Generate service ID
Set creation date and manager ID
Make service available in the catalog
Display success message
View confirmation

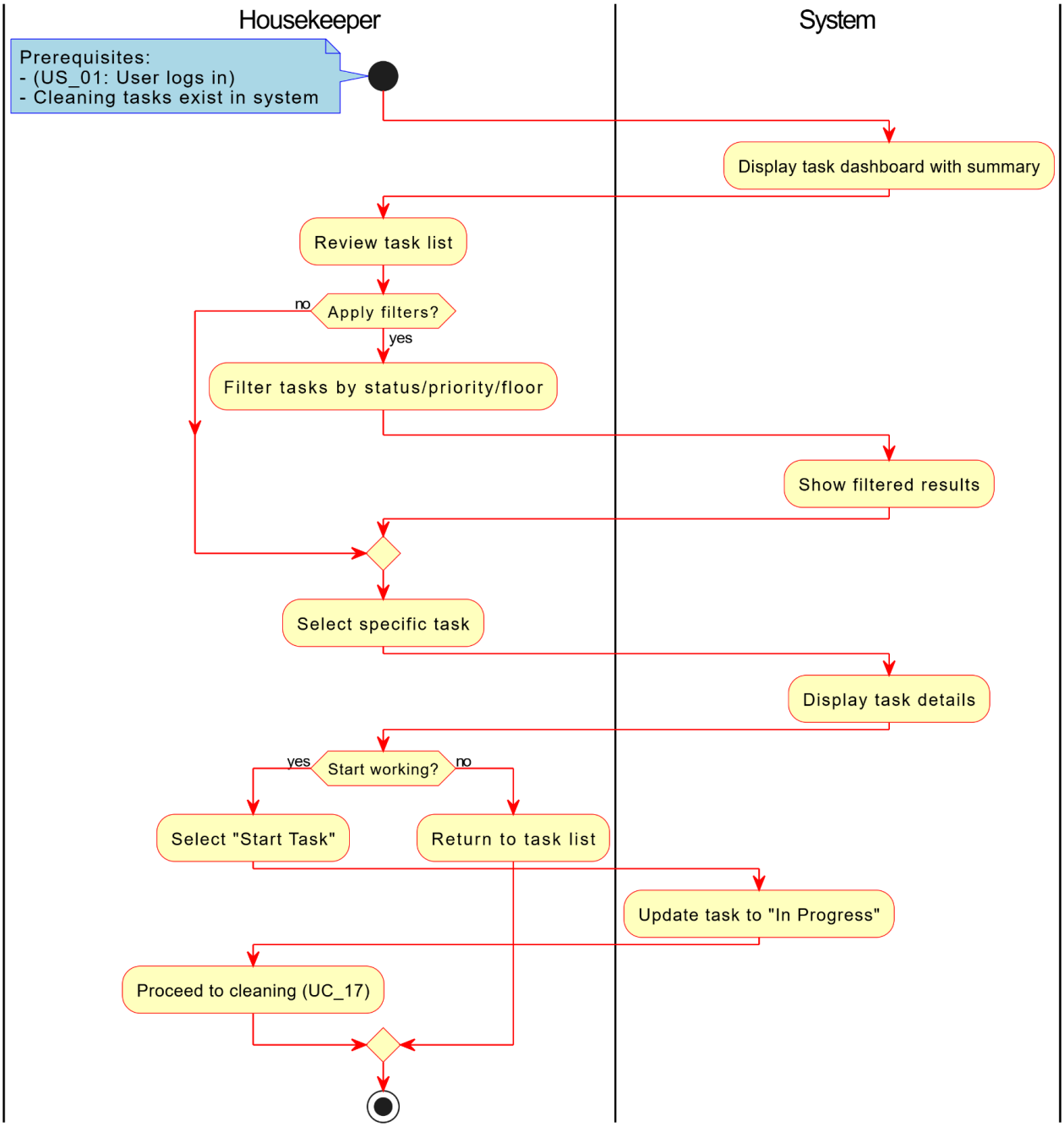

AD_14_Manage_Prices
Manager
System
Prerequisite: Manager is authenticated and has pricing management permissions (See US_01: User logs in)
Navigate to price management section
Manage Room Price or Service Price?
Room Price
Service Price
View list of room types
Select room type to update
View list of services
Select service to update
Display current base price and price history
Display current service price and price history
Regular price update or seasonal pricing?
Regular price update
Seasonal pricing
Enter new base price
Add reason for price change (optional)
Save changes
Select seasonal pricing option
Specify date range for special rate
Enter special rate amount
Add description for seasonal pricing
Save seasonal pricing rule
Validate price input
Price valid?
yes
no
Update price in database
Display validation error
Correct price input
Create price change history record
Apply new pricing to future bookings
Validate price input
Correct invalid inputs
Display success message
Validate date range and price
Input valid?
yes
no
Create seasonal pricing rule
Display validation error
Update pricing calculation logic
Display success message
Validate input
Log price change details:
- Changed by (manager ID)
- Old price
- New price
- Timestamp
- Reason for change
View updated pricing information


AD_15_View_Reservations
Receptionist
System
Prerequisite: Receptionist is authenticated (See US_01: User logs in)
Log in to system
Navigate to reservation management section
View today's arrivals or departures?
Arrivals
Departures
Select "Today's Arrivals" view
Select "Today's Departures" view
Query database for reservations with check-in date = today's date AND status = "Reserved"
Query database for reservations with check-out date = today's date AND status = "CheckedIn"
Any matching reservations?
yes
no
Retrieve reservation details including
Display "No reservations found" message
Display list of reservations sorted by scheduled time
Try different view?
no
yes
Change view type or date range
Review reservation list
Need more details?
no
yes
Select specific reservation
Display detailed reservation information
Review detailed information
Apply custom filter?
no
yes
Select filter options
Enter custom date range if needed
Apply filters
Requery database with filter criteria
Update displayed reservations
Perform action on reservation?
no
yes
Arrival or departure?
Arrival
Departure
Select reservation for check-in
Select reservation for check-out
Proceed to check-in process
Proceed to check-out process
Exit reservation view

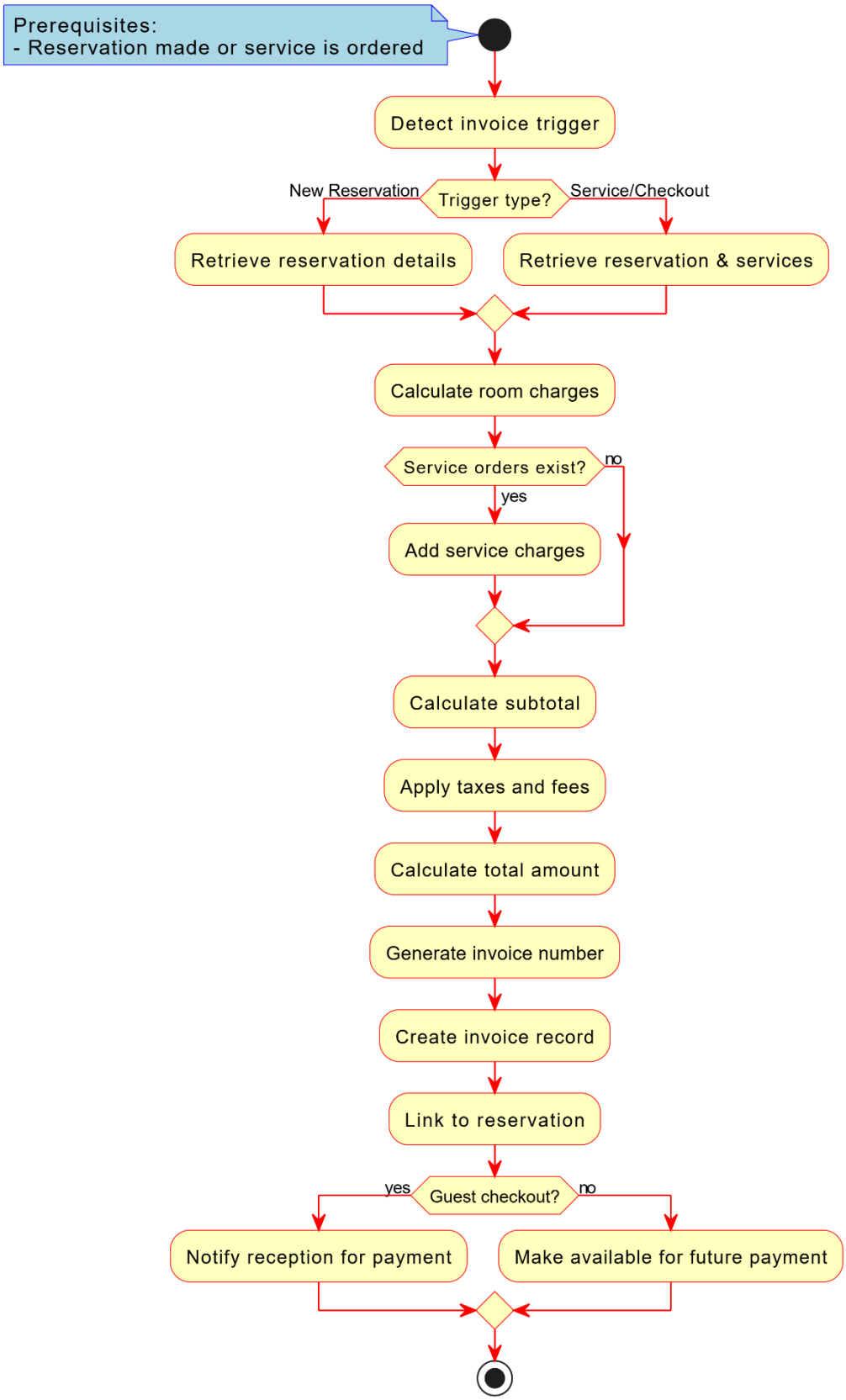

AD_16_Report_Issue
Guest
System
Prerequisites:
- Guest is authenticated (US_01: User logs in)
- Guest has active reservation
- Guest is currently staying at hotel
Navigate to support/help section
Select "Report Issue" option
Display issue reporting form
Pre-populate room information based on guest's current reservation
Select issue category (Room Problem, Maintenance, Service Issue, etc.)
Enter issue description
Select priority/urgency level:
- Low
- Medium
- High
Add photo evidence?
no
yes
Take photo or select from device
Upload photo
Validate and process photo
Review issue details
Submit issue report
Validate submission
Validation successful?
yes
no
Save issue report to database
Display validation errors
Create maintenance request with status "Reported"
Save draft of issue report
Correct errors in submission
Priority is high?
yes
no
Flag issue as urgent
Add to regular maintenance queue
Resubmit issue report
Issue relates to room?
no
yes
Update room status to "Maintenance" if not occupied
Send immediate notification to maintenance staff
Generate confirmation for guest
Notify appropriate staff based on issue category
View confirmation message with:
- Issue tracking number
- Estimated response time
- Contact information
System error occurs?
no
yes
Log error details
Save draft of issue report
Display error message to guest
Retry submission or contact front desk
Exit issue reporting system

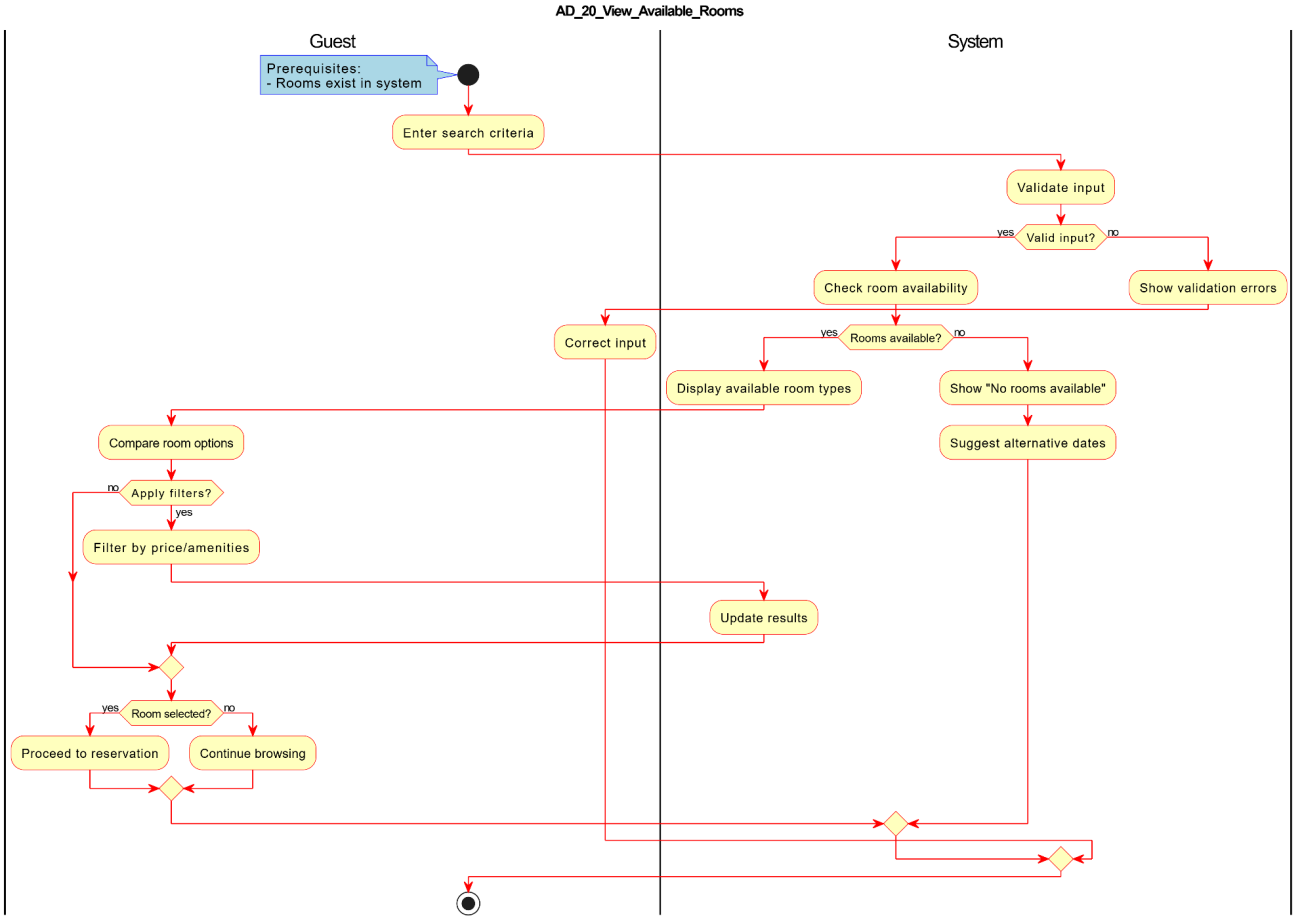

AD_17_Update_Cleaning_Status
Housekeeper
System
Prerequisites:
- Housekeeper is authenticated (US_01: User logs in)
- Room is assigned for cleaning
- Room status is "Dirty"
Navigate to assigned cleaning tasks
View list of assigned rooms
Select specific room to clean
Display detailed cleaning task information
Mark task as "In Progress"
Update cleaning task status to "InProgress"
Log task started timestamp
Physically perform room cleaning
Inspect room for cleanliness
Any maintenance issues found?
no
yes
Document maintenance issue
Take photo of issue (optional)
Select "Report Maintenance Issue" option
Display maintenance reporting form
Select issue category
Enter issue description
Select priority level
Submit maintenance report
Create maintenance request with status "Reported"
Assign maintenance request to appropriate staff
Link maintenance request to room
Mark room as "Cleaned"
Add cleaning notes (optional)
Submit cleaning status update
Validate cleaning status update
Validation successful?
yes
no
Update cleaning task status to "Cleaned"
Display error message
Review and correct issues
Record completion timestamp
Update room status to "Available"
Update room's "NeedsCleaning" flag to false
Update room's "LastCleaned" timestamp
Record housekeeper ID as "CleanedById"
Maintenance issue reported?
yes
no
Keep room status as "Maintenance"
Make room available for new reservations
Notify maintenance staff
Notify reception that room is ready
Display confirmation message
Return to task list

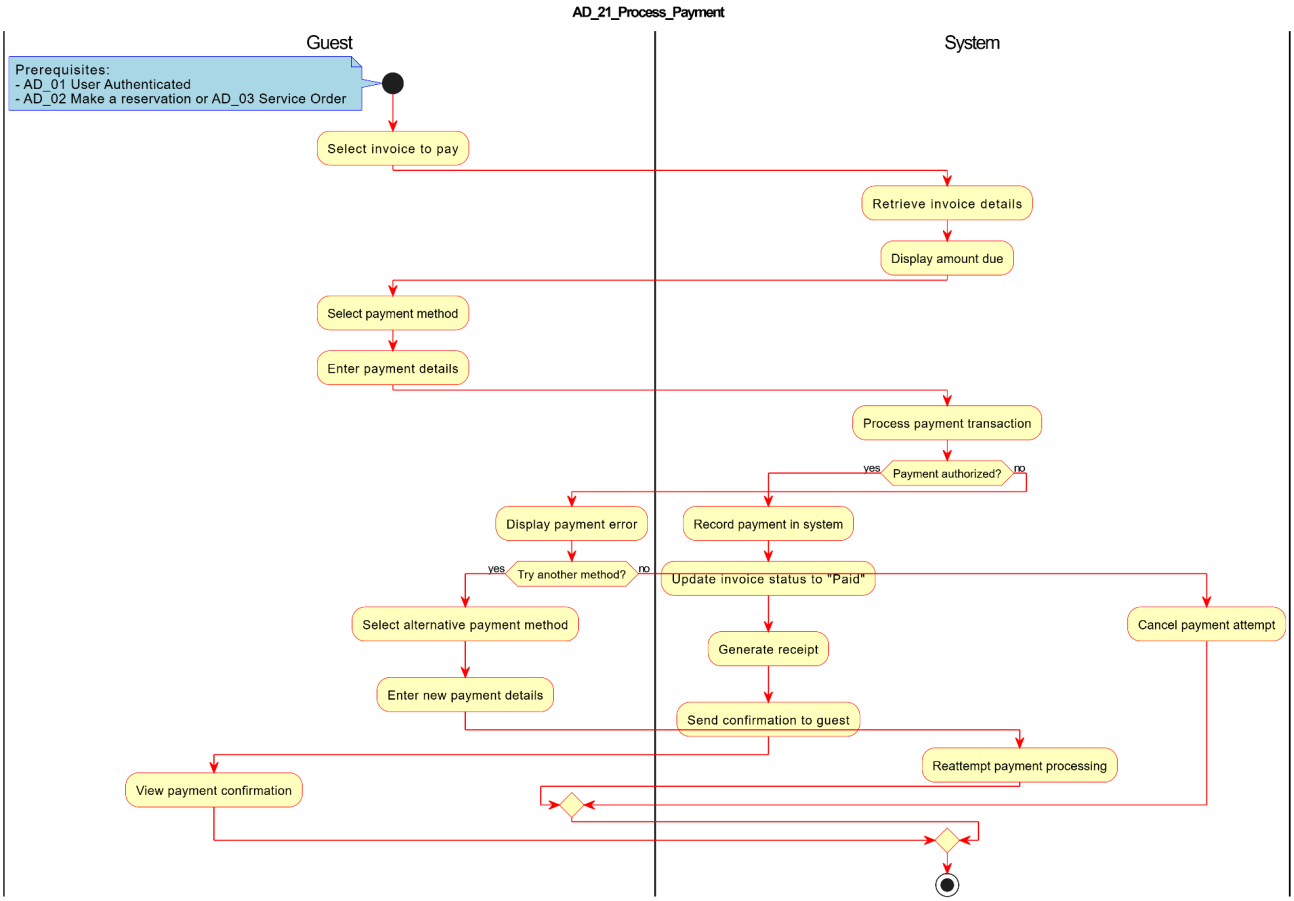
AD_18_View_Cleaning_Tasks

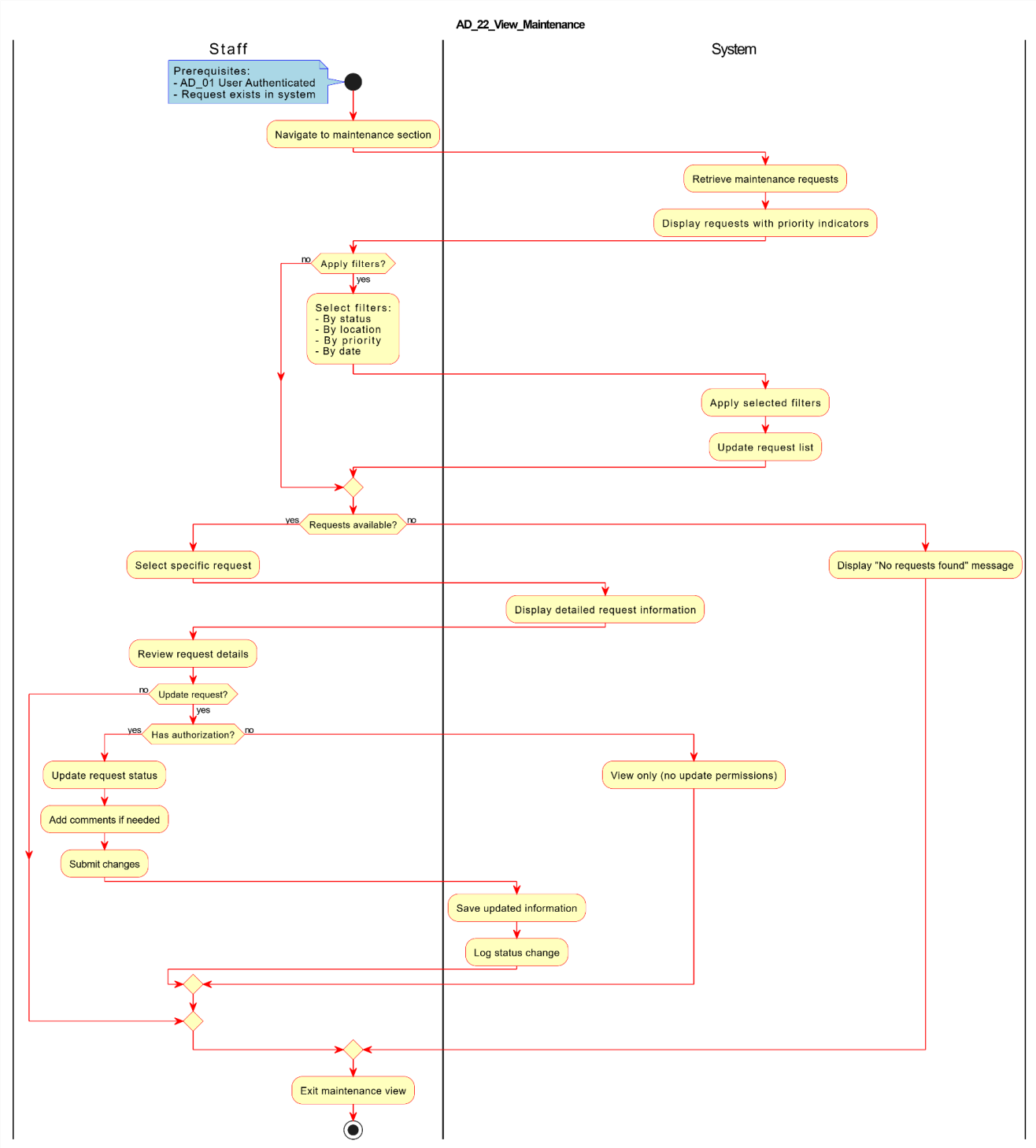


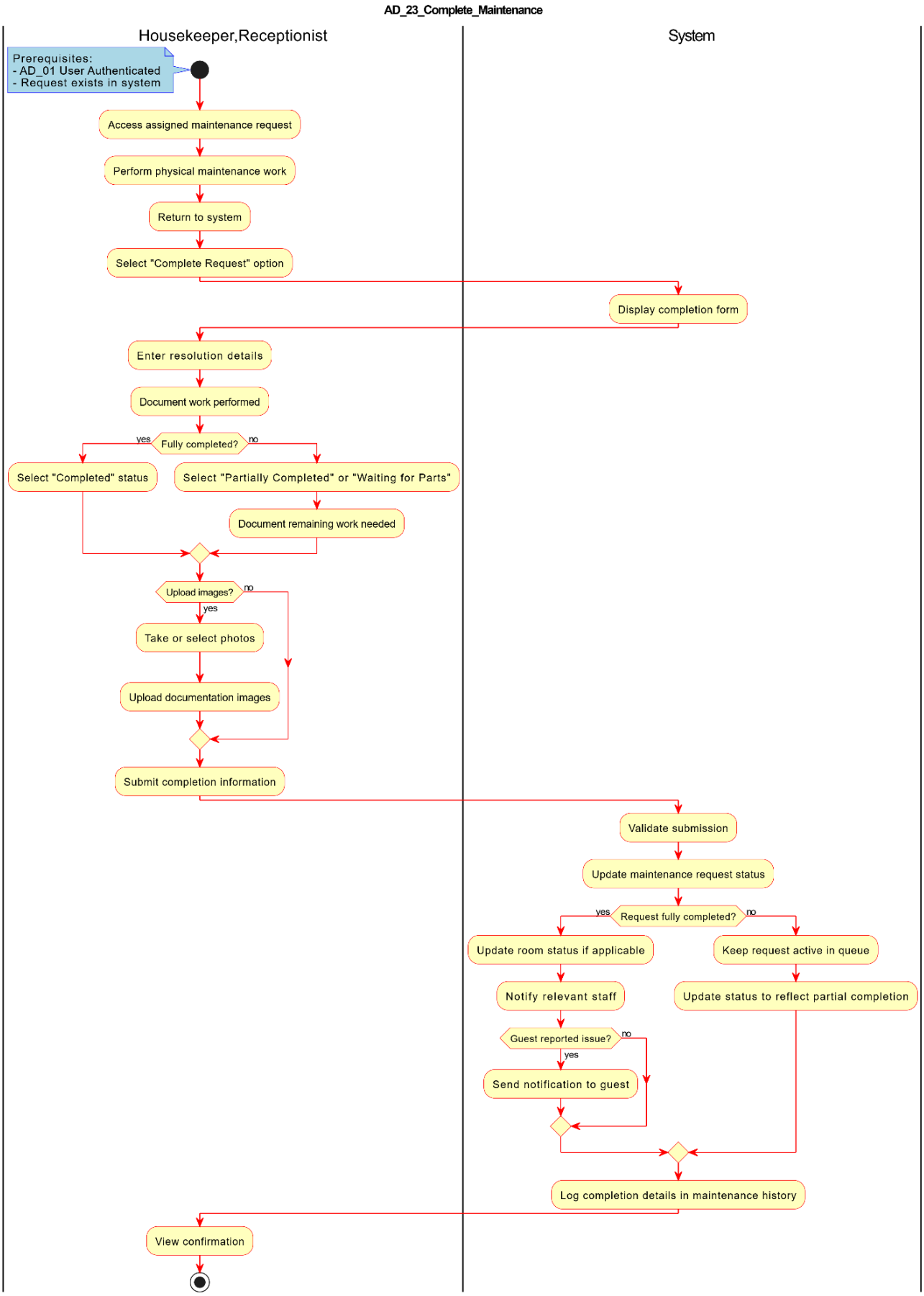
AD_19_Generate_Invoice

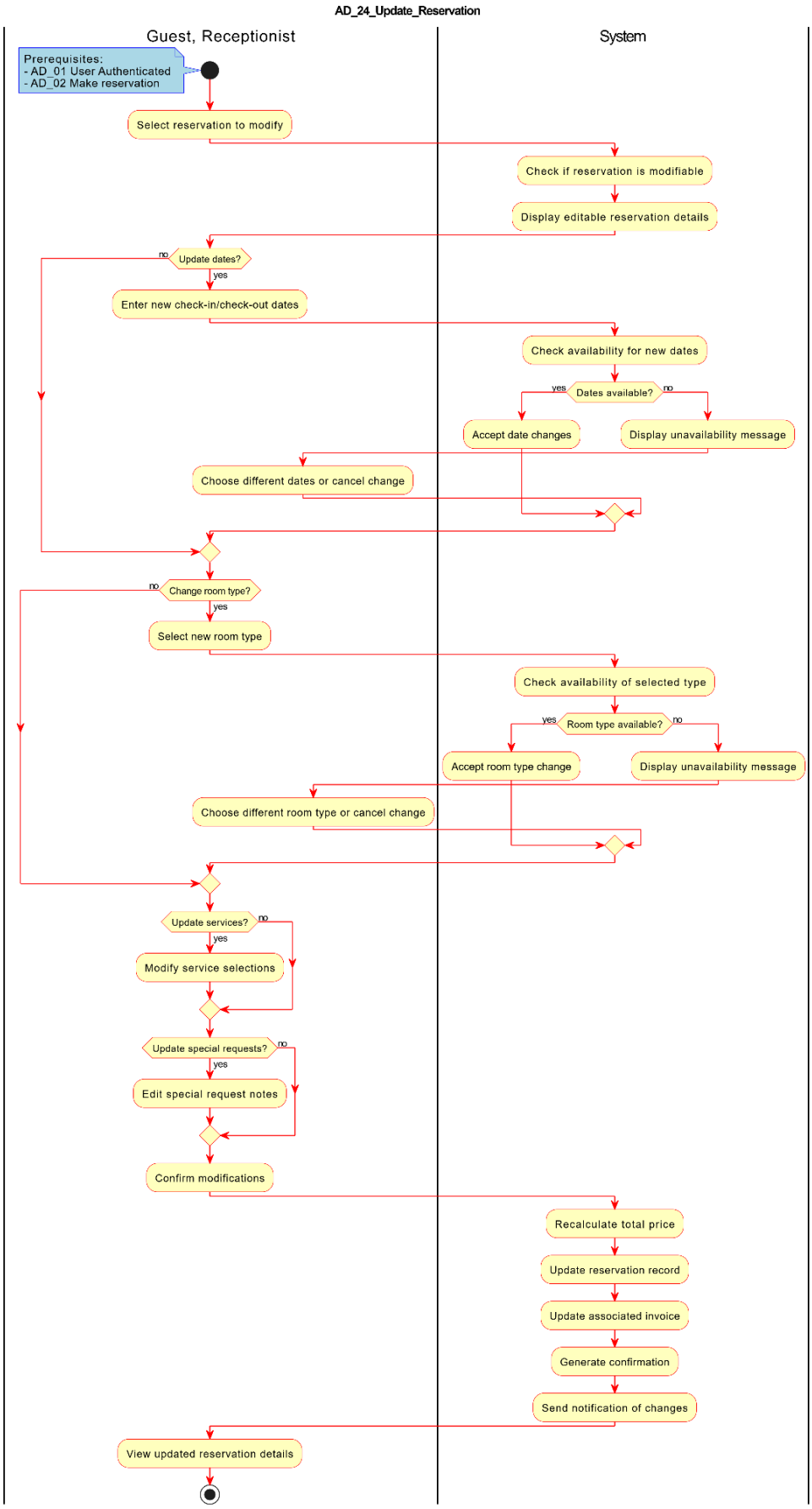


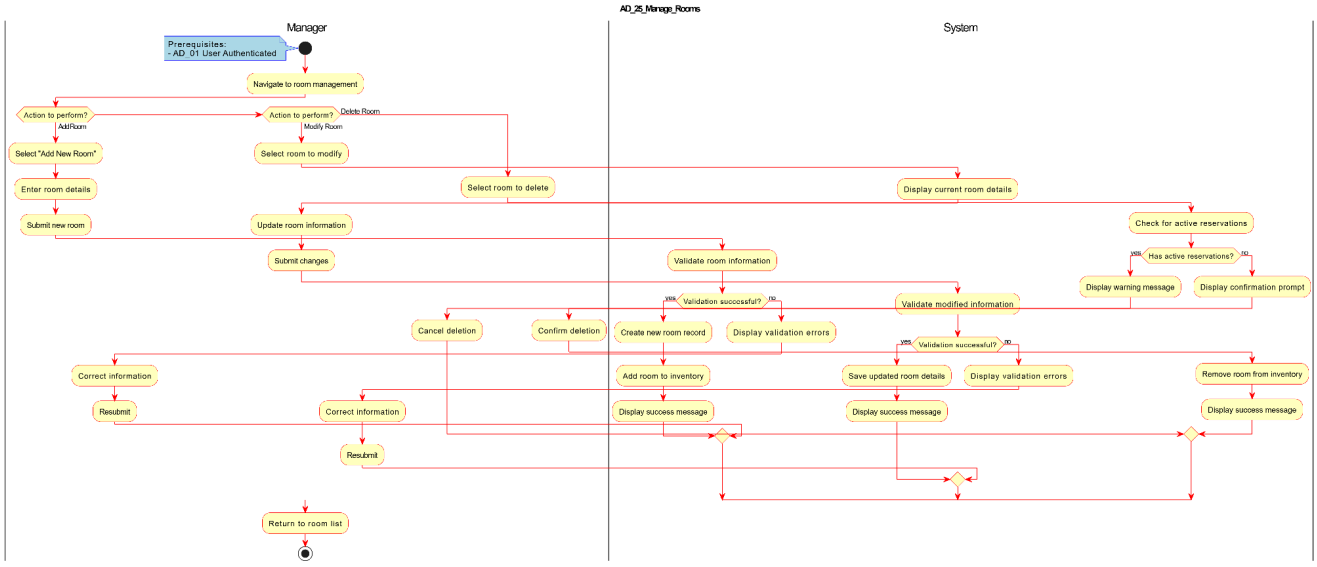

AD_20_View_Available_Rooms
Guest
System
Prerequisites:
- Rooms exist in system
Enter search criteria
Validate input
Valid input?
yes
no
Check room availability
Show validation errors
Correct input
Rooms available?
yes
no
Display available room types
Show "No rooms available"
Suggest alternative dates
Compare room options
Apply filters?
no
yes
Filter by price/amenities
Update results
Room selected?
yes
no
Proceed to reservation
Continue browsing


AD_21_Process_Payment
Guest
System
Prerequisites:
- AD_01 User Authenticated
- AD_02 Make a reservation or AD_03 Service Order
Select invoice to pay
Retrieve invoice details
Display amount due
Select payment method
Enter payment details
Process payment transaction
yes
Payment authorized?
no
Display payment error
Record payment in system
yes
Try another method?
no
Update invoice status to "Paid"
Select alternative payment method
Cancel payment attempt
Generate receipt
Enter new payment details
Send confirmation to guest
Reattempt payment processing
View payment confirmation

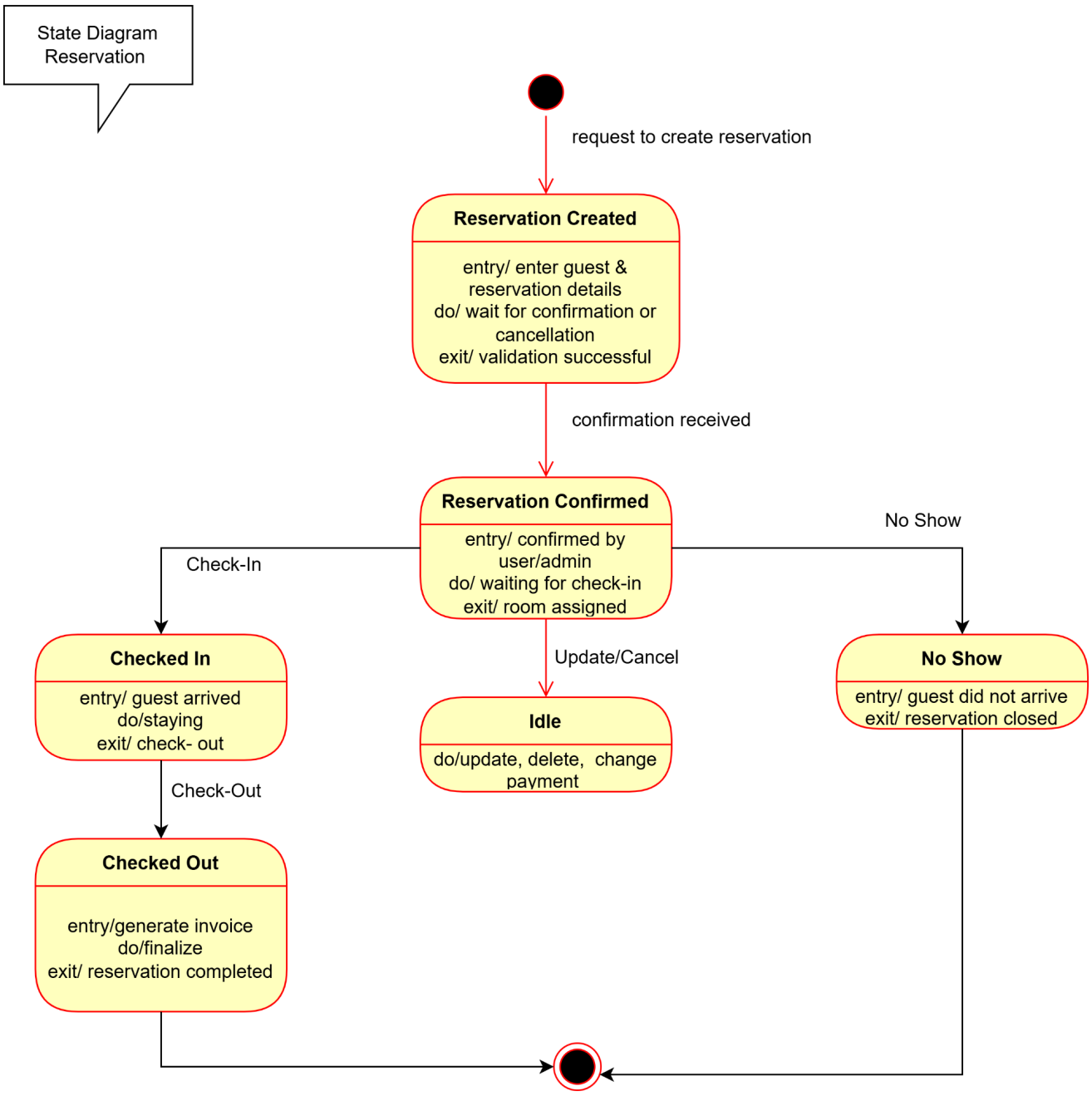

AD_22_View_Maintenance
Staff
System
Prerequisites:
- AD_01 User Authenticated
- Request exists in system
Navigate to maintenance section
Retrieve maintenance requests
Display requests with priority indicators
Apply filters?
no
yes
Select filters:
- By status
- By location
- By priority
- By date
Apply selected filters
Update request list
Requests available?
yes
no
Select specific request
Display "No requests found" message
Display detailed request information
Review request details
Update request?
no
yes
Has authorization?
yes
no
Update request status
View only (no update permissions)
Add comments if needed
Submit changes
Save updated information
Log status change
Exit maintenance view

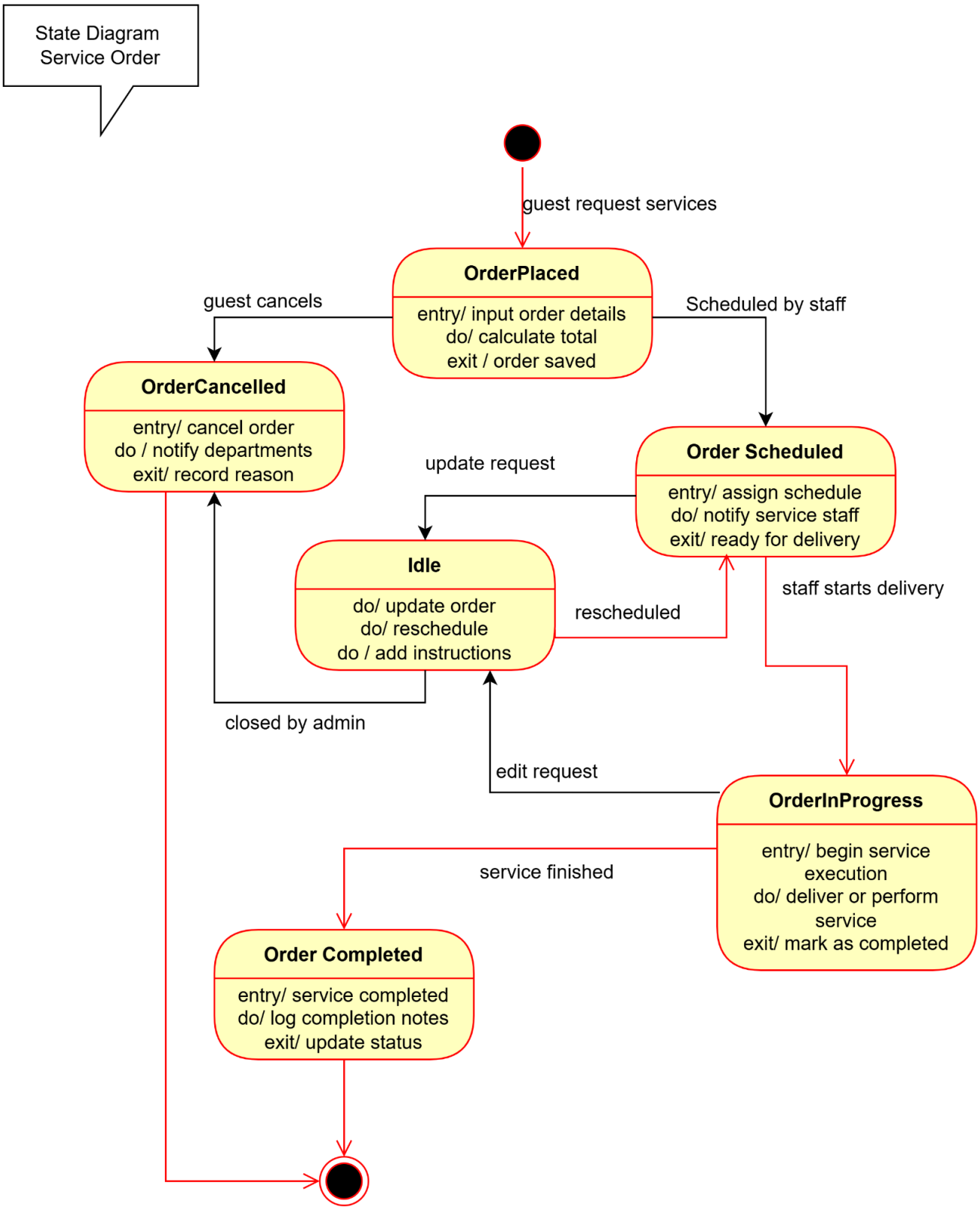

AD_23_Complete_Maintenance
Housekeeper,Receptionist
System
Prerequisites:
- AD_01 User Authenticated
- Request exists in system
Access assigned maintenance request
Perform physical maintenance work
Return to system
Select "Complete Request" option
Display completion form
Enter resolution details
Document work performed
Fully completed?
yes
no
Select "Completed" status
Select "Partially Completed" or "Waiting for Parts"
Document remaining work needed
Upload images?
yes
no
Take or select photos
Upload documentation images
Submit completion information
Validate submission
Update maintenance request status
Request fully completed?
yes
no
Update room status if applicable
Keep request active in queue
Notify relevant staff
Update status to reflect partial completion
Guest reported issue?
yes
no
Send notification to guest
Log completion details in maintenance history
View confirmation

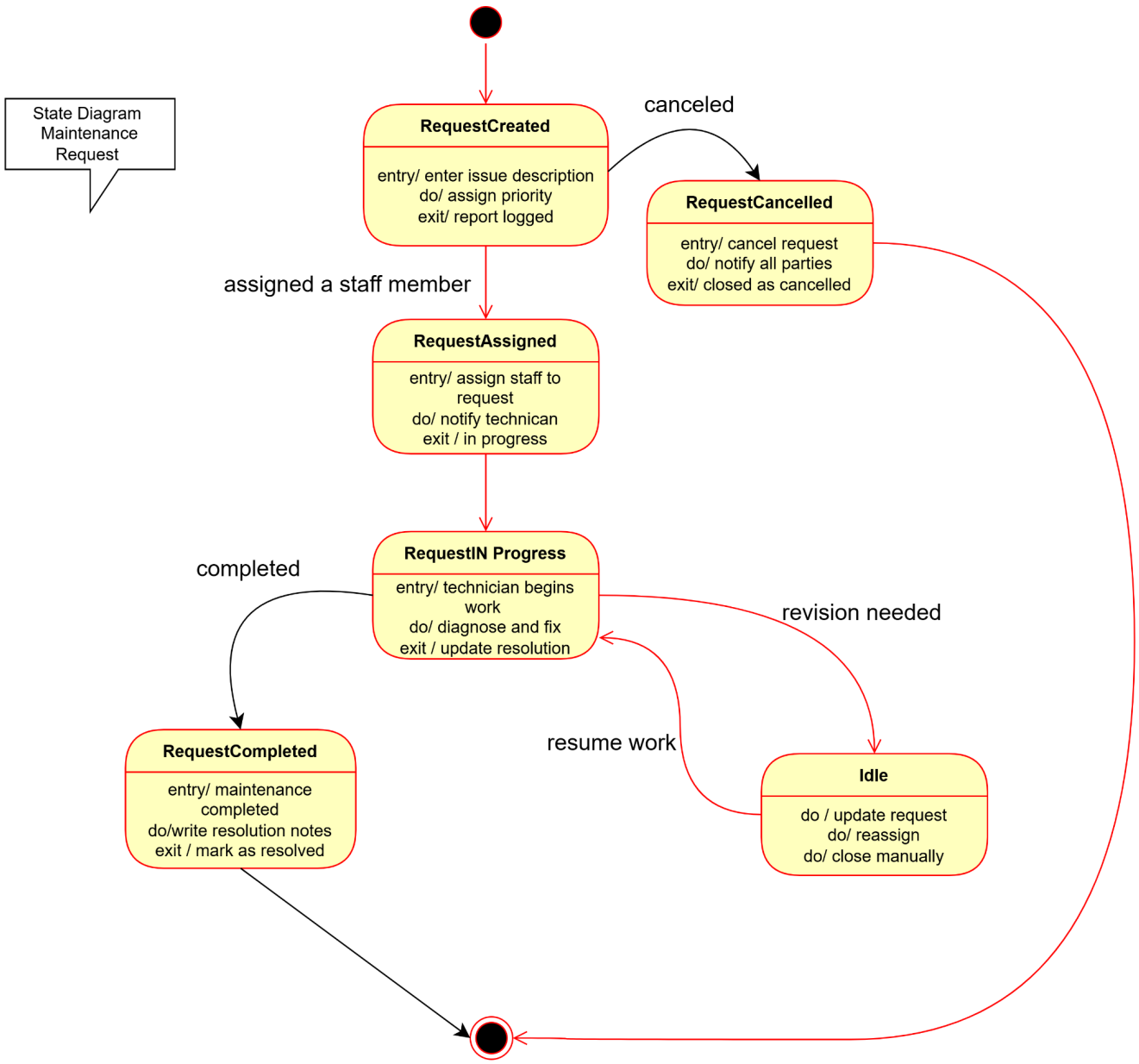

AD_24_Update_Reservation
Guest, Receptionist
System
Prerequisites:
- AD_01 User Authenticated
- AD_02 Make reservation
Select reservation to modify
Check if reservation is modifiable
Display editable reservation details
Update dates?
no
yes
Enter new check-in/check-out dates
Check availability for new dates
Dates available?
yes
no
Accept date changes
Display unavailability message
Choose different dates or cancel change
Change room type?
no
yes
Select new room type
Check availability of selected type
Room type available?
yes
no
Accept room type change
Display unavailability message
Choose different room type or cancel change
Update services?
no
yes
Modify service selections
Update special requests?
no
yes
Edit special request notes
Confirm modifications
Recalculate total price
Update reservation record
Update associated invoice
Generate confirmation
Send notification of changes
View updated reservation details

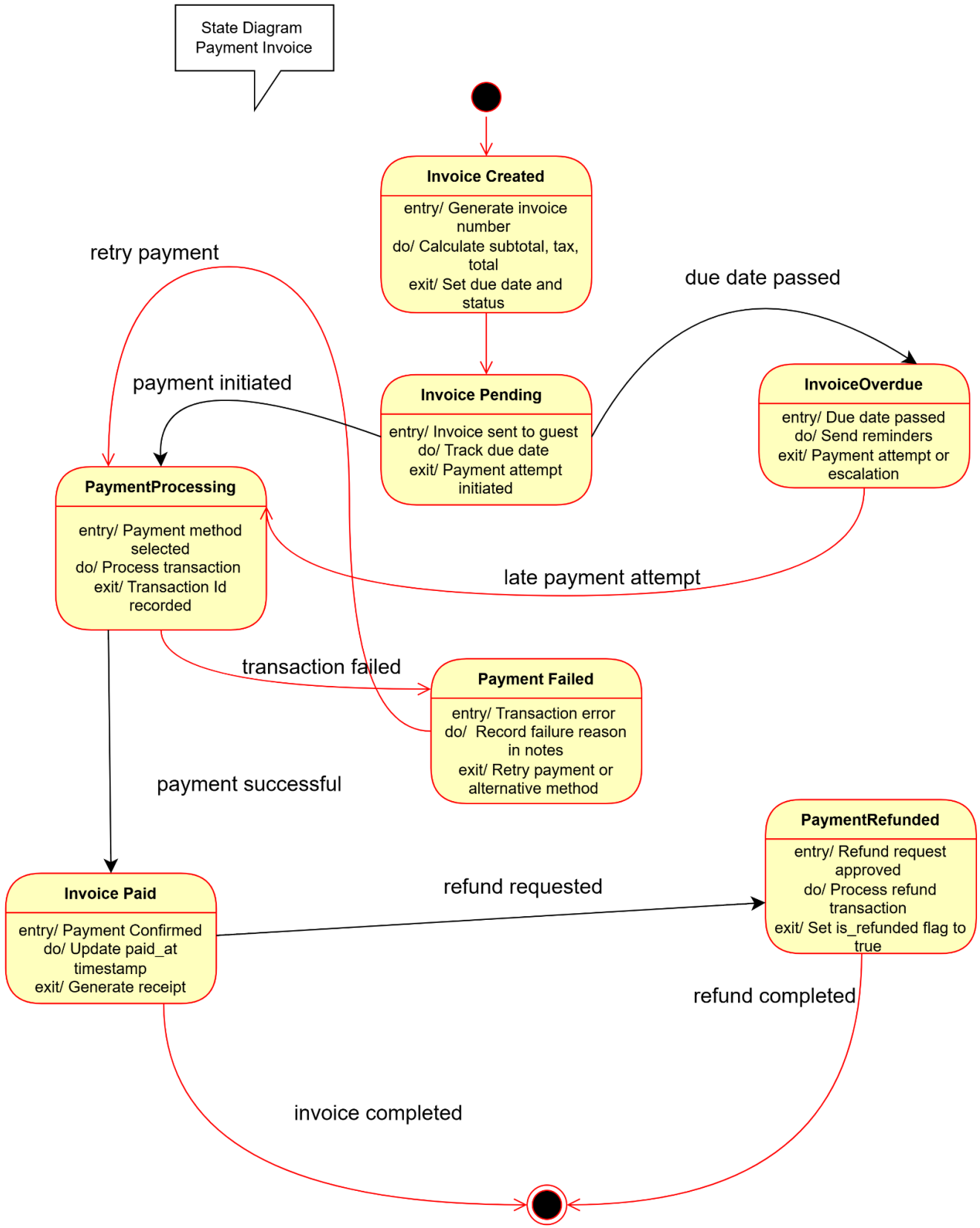

AD_25_Manage_Rooms
Manager
System
Prerequisites:
- AD_01 User Authenticated
Navigate to room management
Action to perform?
Add Room
Action to perform?
Delete Room
Modify Room
Select "Add New Room"
Enter room details
Submit new room
Select room to modify
Update room information
Submit changes
Select room to delete
Display current room details
Check for active reservations
Has active reservations?
yes
no
Display warning message
Display confirmation prompt
Validate room information
Validation successful?
yes
no
Create new room record
Display validation errors
Add room to inventory
Display success message
Correct information
Resubmit
Validate modified information
Validation successful?
yes
no
Save updated room details
Display validation errors
Display success message
Cancel deletion
Confirm deletion
Correct information
Resubmit
Remove room from inventory
Display success message
Return to room list

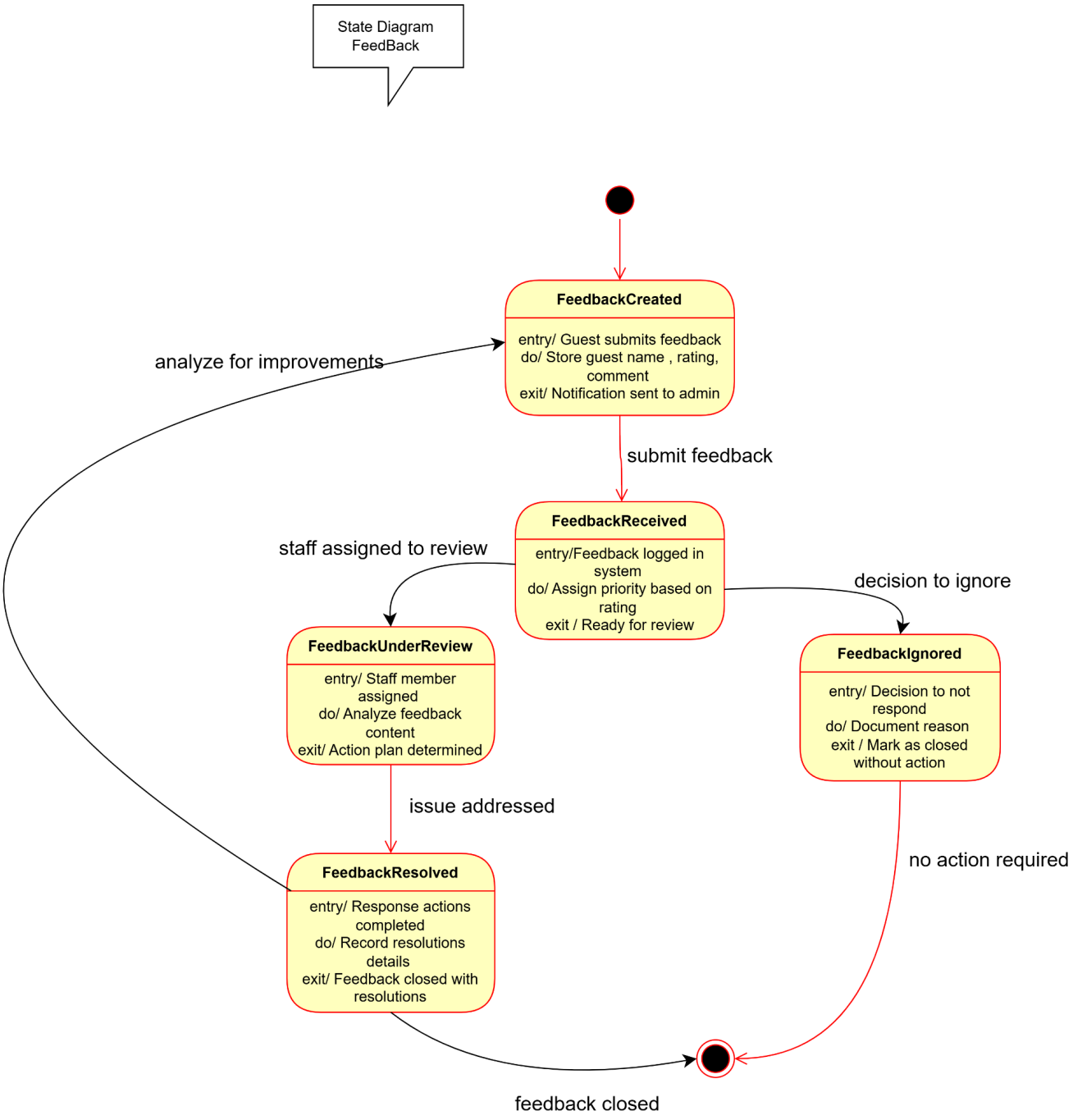
4.2.3 State Diagrams




State Diagram
Service Order
guest request services
OrderPlaced
entry/ input order details
do/ calculate total
exit / order saved
guest cancels
Scheduled by staff
OrderCancelled
entry/ cancel order
do / notify departments
exit/ record reason
Order Scheduled
entry/ assign schedule
do/ notify service staff
exit/ ready for delivery
update request
Idle
do/ update order
do/ reschedule
do / add instructions
rescheduled
staff starts delivery
closed by admin
edit request
OrderInProgress
entry/ begin service execution
do/ deliver or perform service
exit/ mark as completed
service finished
Order Completed
entry/ service completed
do/ log completion notes
exit/ update status

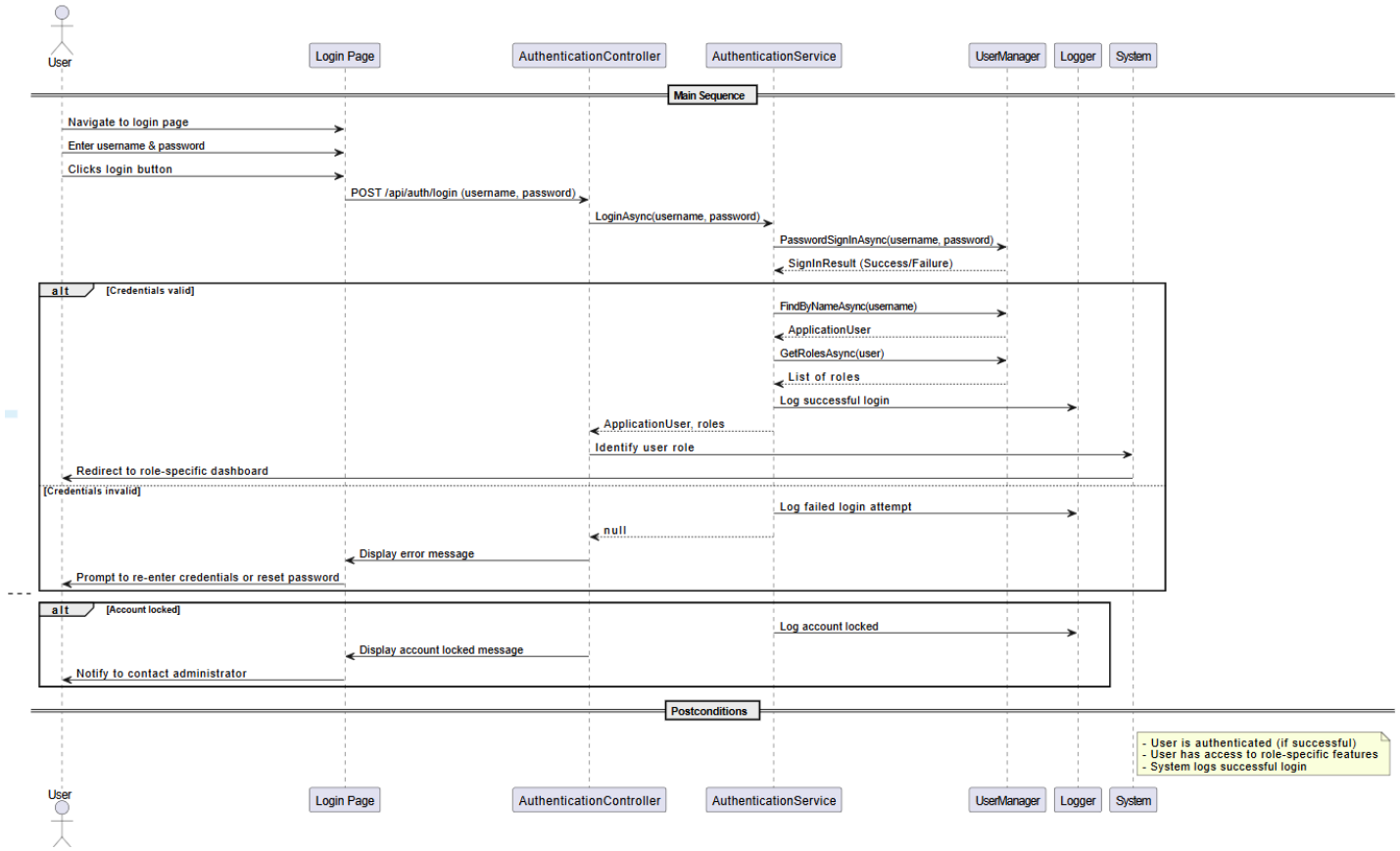

State Diagram
Maintenance
Request
RequestCreated
entry/ enter issue description
do/ assign priority
exit/ report logged
canceled
RequestCancelled
entry/ cancel request
do/ notify all parties
exit/ closed as cancelled
assigned a staff member
RequestAssigned
entry/ assign staff to
request
do/ notify technican
exit / in progress
RequestIN Progress
entry/ technician begins
work
do/ diagnose and fix
exit / update resolution
completed
revision needed
resume work
RequestCompleted
entry/ maintenance
completed
do/write resolution notes
exit / mark as resolved
Idle
do / update request
do/ reassign
do/ close manually


State Diagram
Payment Invoice
Invoice Created
entry/ Generate invoice number
do/ Calculate subtotal, tax, total
exit/ Set due date and status
retry payment
due date passed
payment initiated
Invoice Pending
entry/ Invoice sent to guest
do/ Track due date
exit/ Payment attempt initiated
InvoiceOverdue
entry/ Due date passed
do/ Send reminders
exit/ Payment attempt or escalation
PaymentProcessing
entry/ Payment method selected
do/ Process transaction
exit/ Transaction Id recorded
late payment attempt
transaction failed
Payment Failed
entry/ Transaction error
do/ Record failure reason in notes
exit/ Retry payment or alternative method
payment successful
PaymentRefunded
entry/ Refund request approved
do/ Process refund transaction
exit/ Set is_refunded flag to true
Invoice Paid
entry/ Payment Confirmed
do/ Update paid_at timestamp
exit/ Generate receipt
refund requested
refund completed
invoice completed

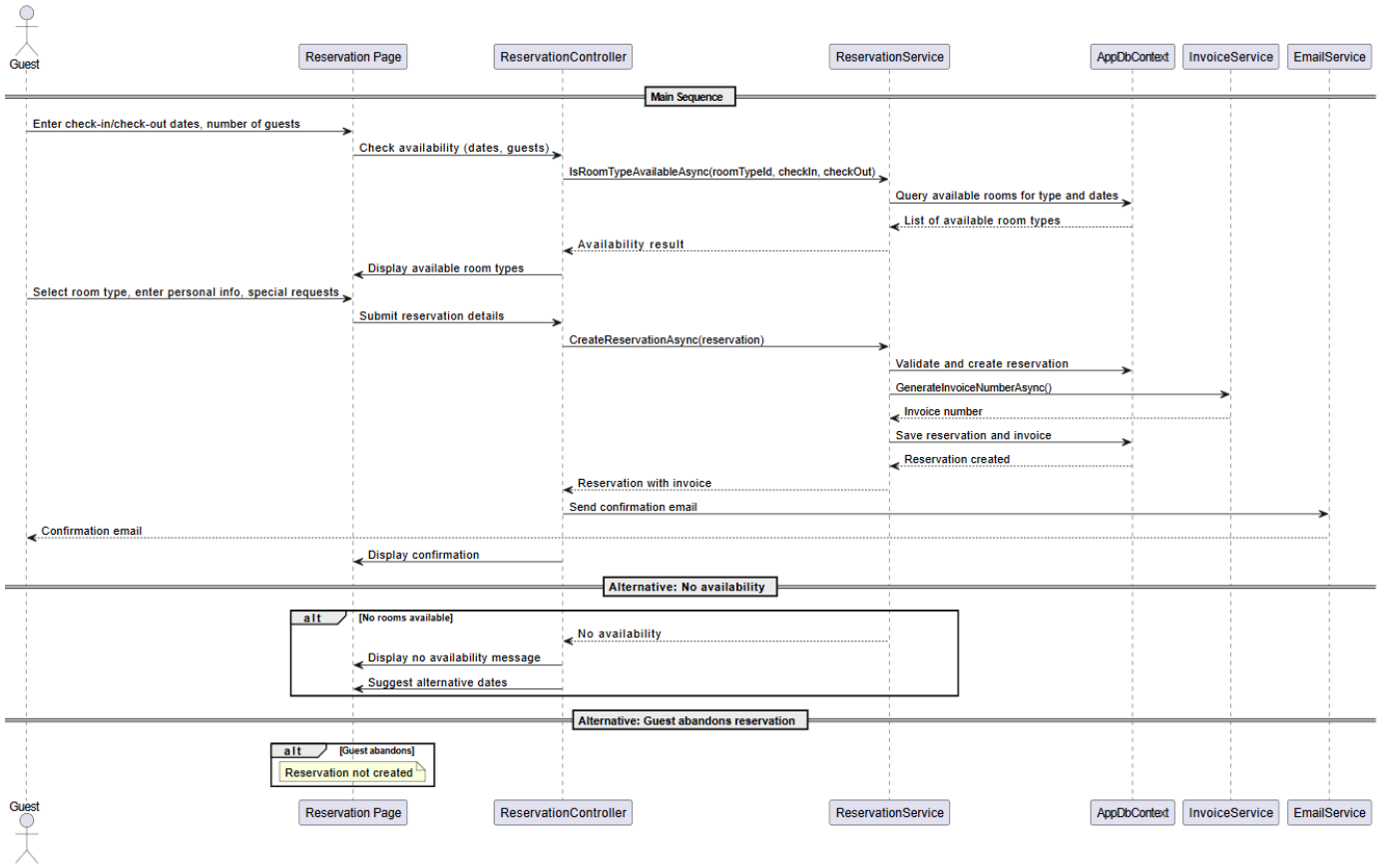

State Diagram
FeedBack
FeedbackCreated
entry/ Guest submits feedback
do/ Store guest name , rating, comment
exit/ Notification sent to admin
submit feedback
FeedbackReceived
entry/Feedback logged in system
do/ Assign priority based on rating
exit / Ready for review
staff assigned to review
decision to ignore
FeedbackUnderReview
entry/ Staff member assigned
do/ Analyze feedback content
exit/ Action plan determined
FeedbackIgnored
entry/ Decision to not respond
do/ Document reason
exit / Mark as closed without action
issue addressed
no action required
FeedbackResolved
entry/ Response actions completed
do/ Record resolutions details
exit/ Feedback closed with resolutions
analyze for improvements
feedback closed

4.2.4 Sequence Diagrams

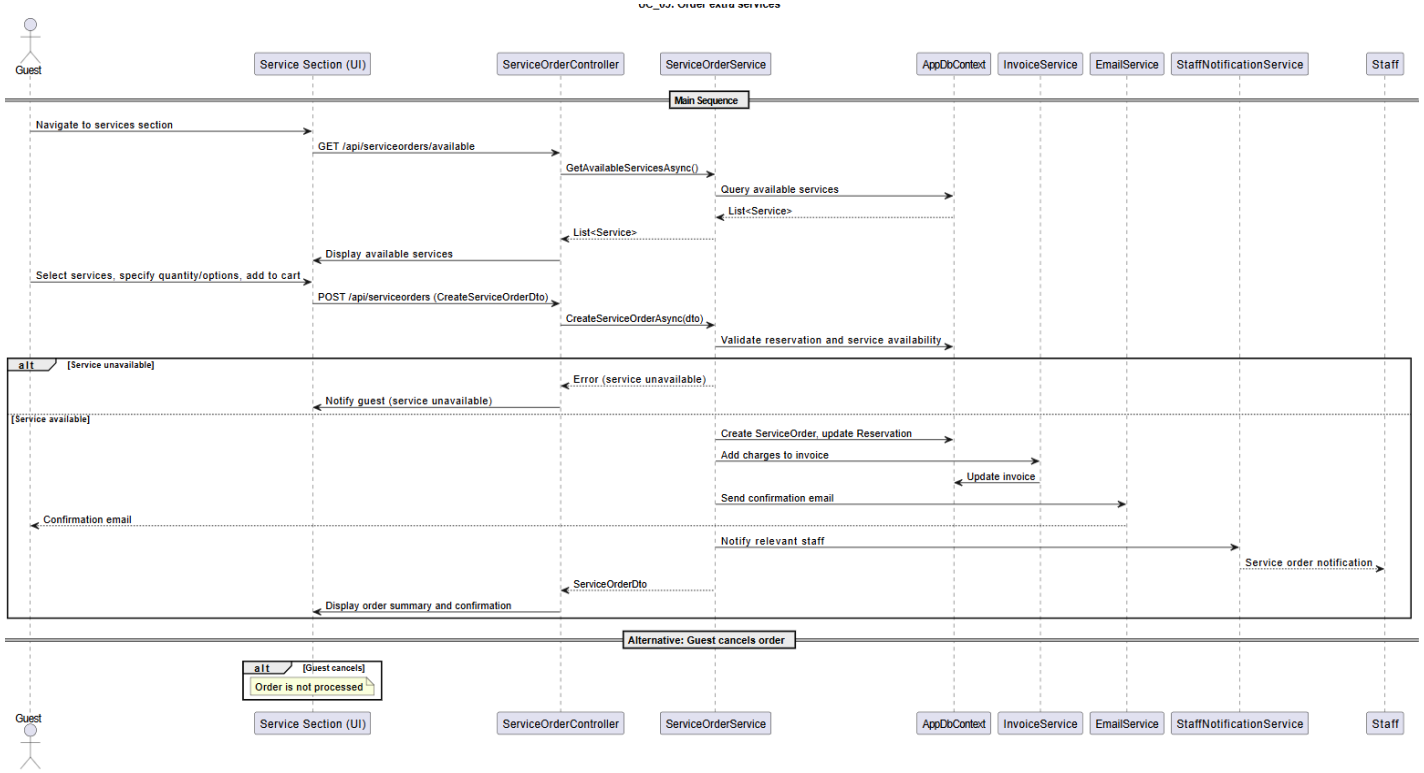
SeqD_1 - User Logs in



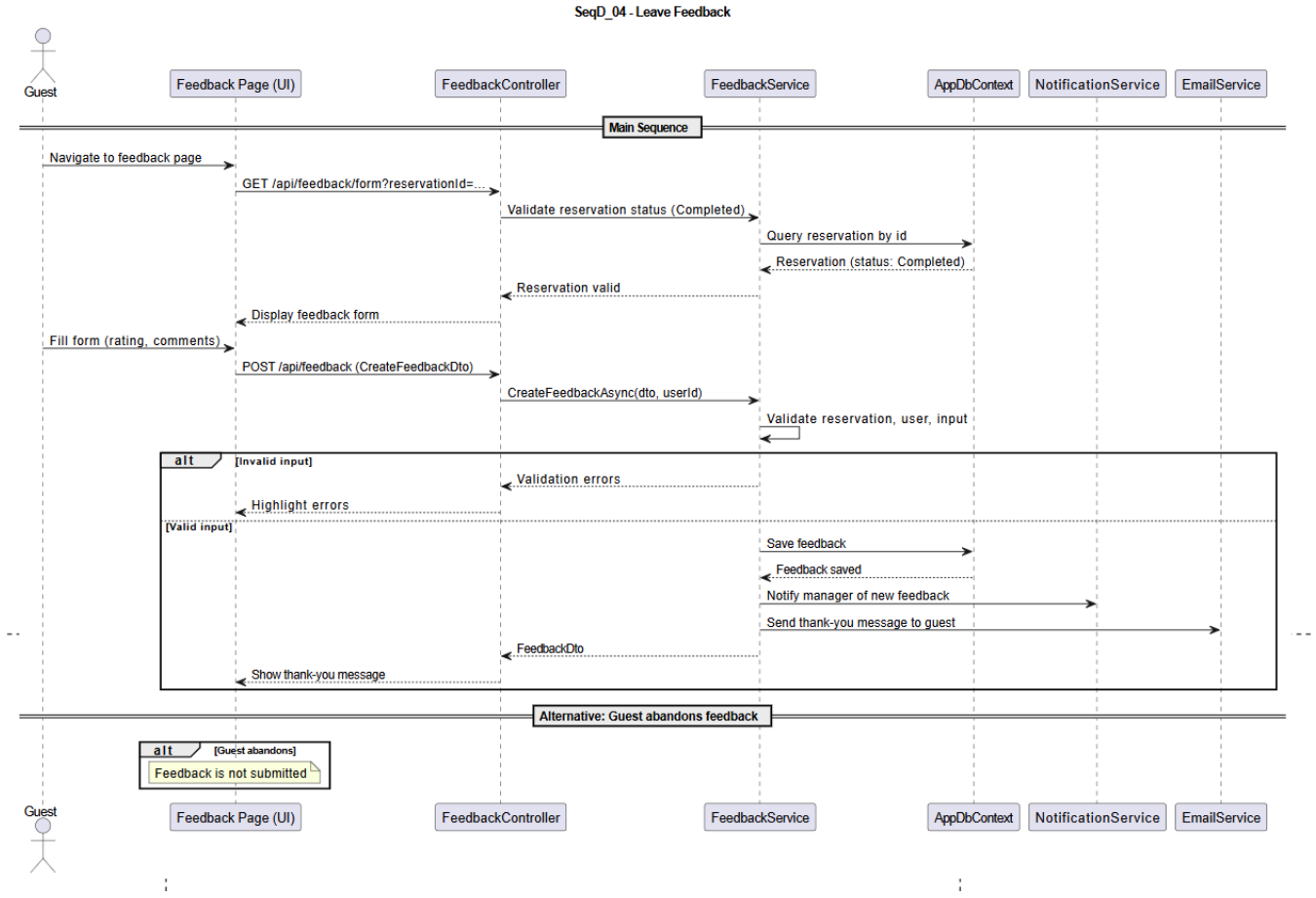
SeqD_2 - Make a reservation



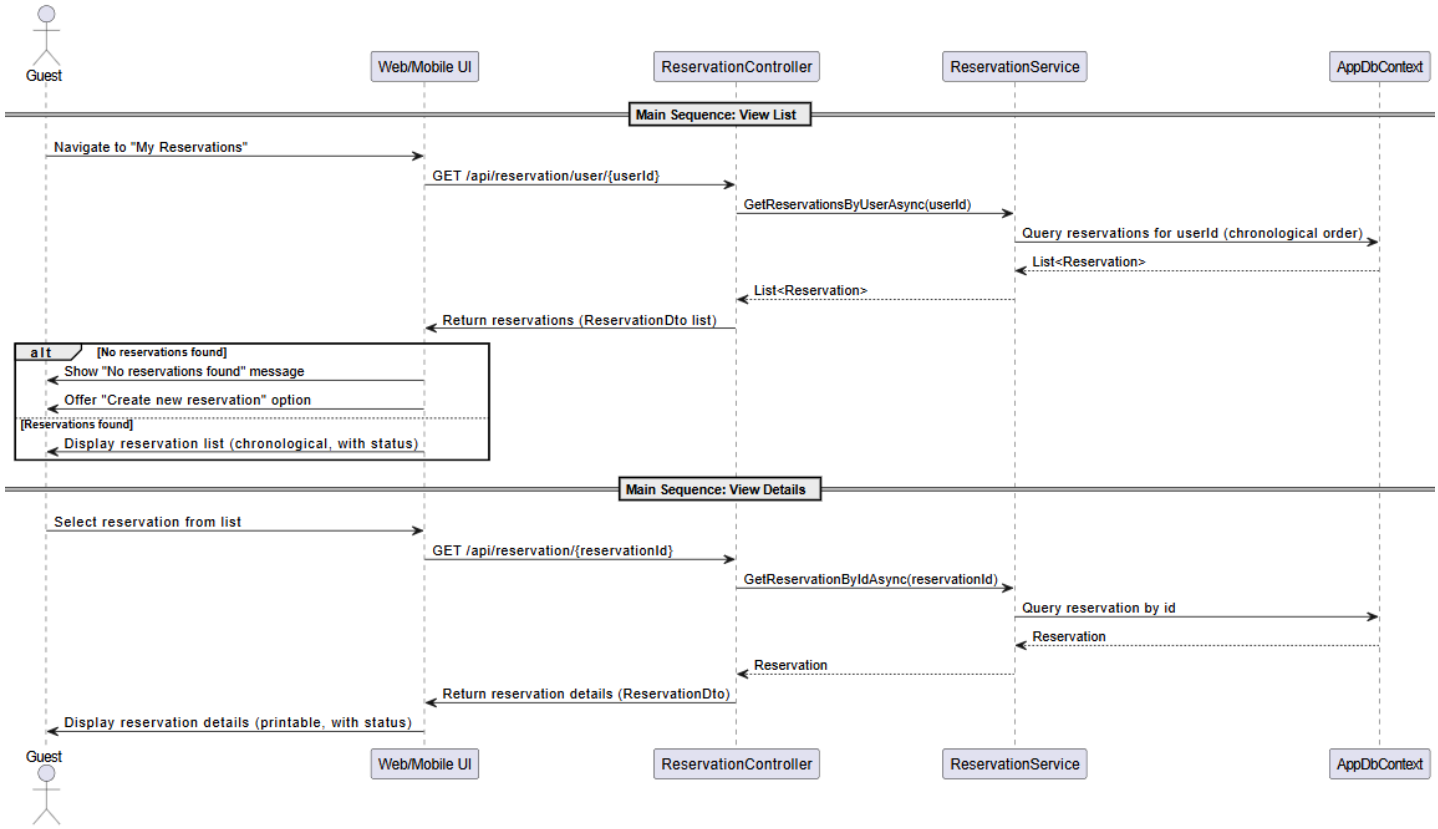
SeqD_3 -Order extra services



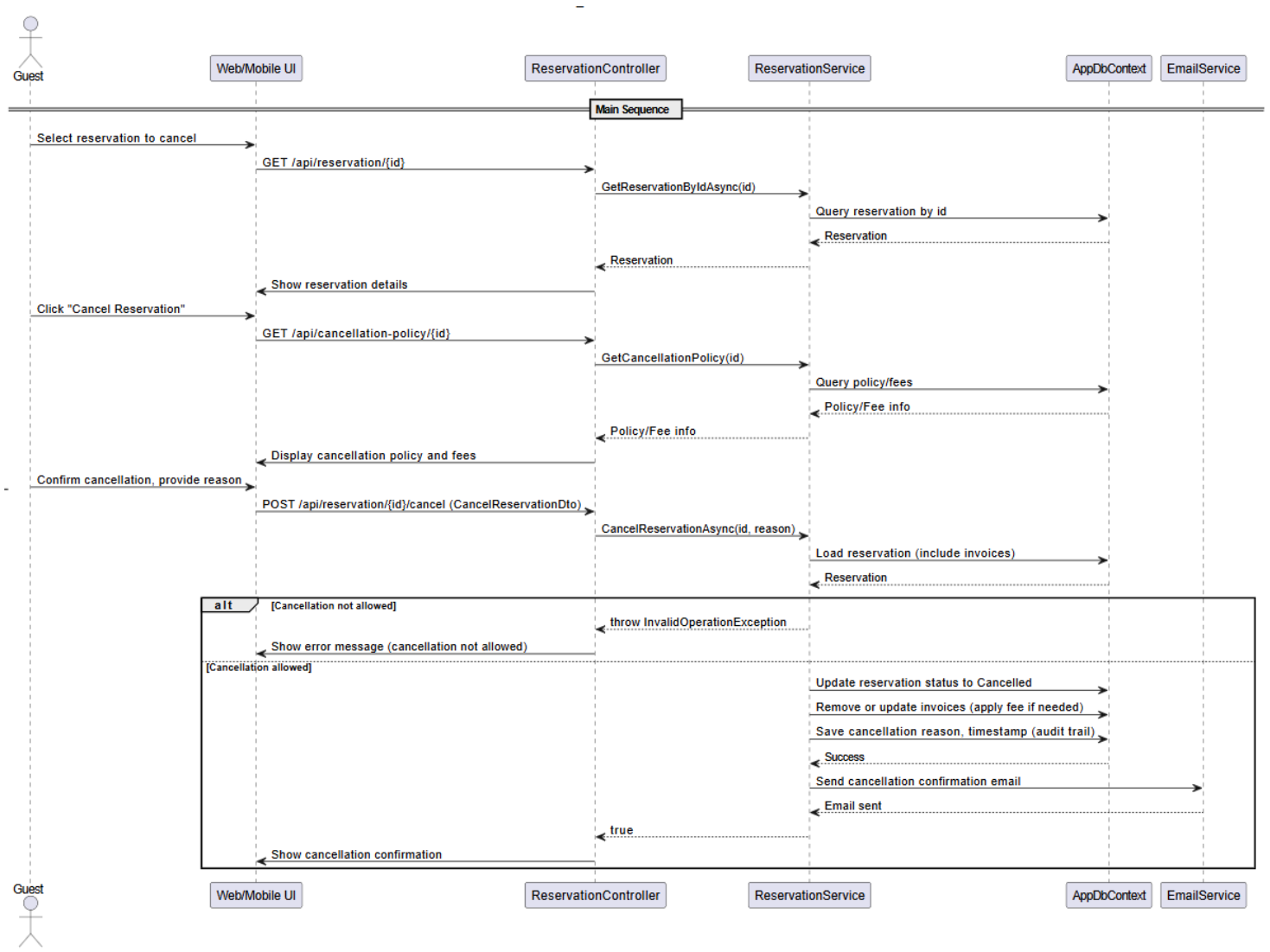
SeqD_04 - Leave Feedback



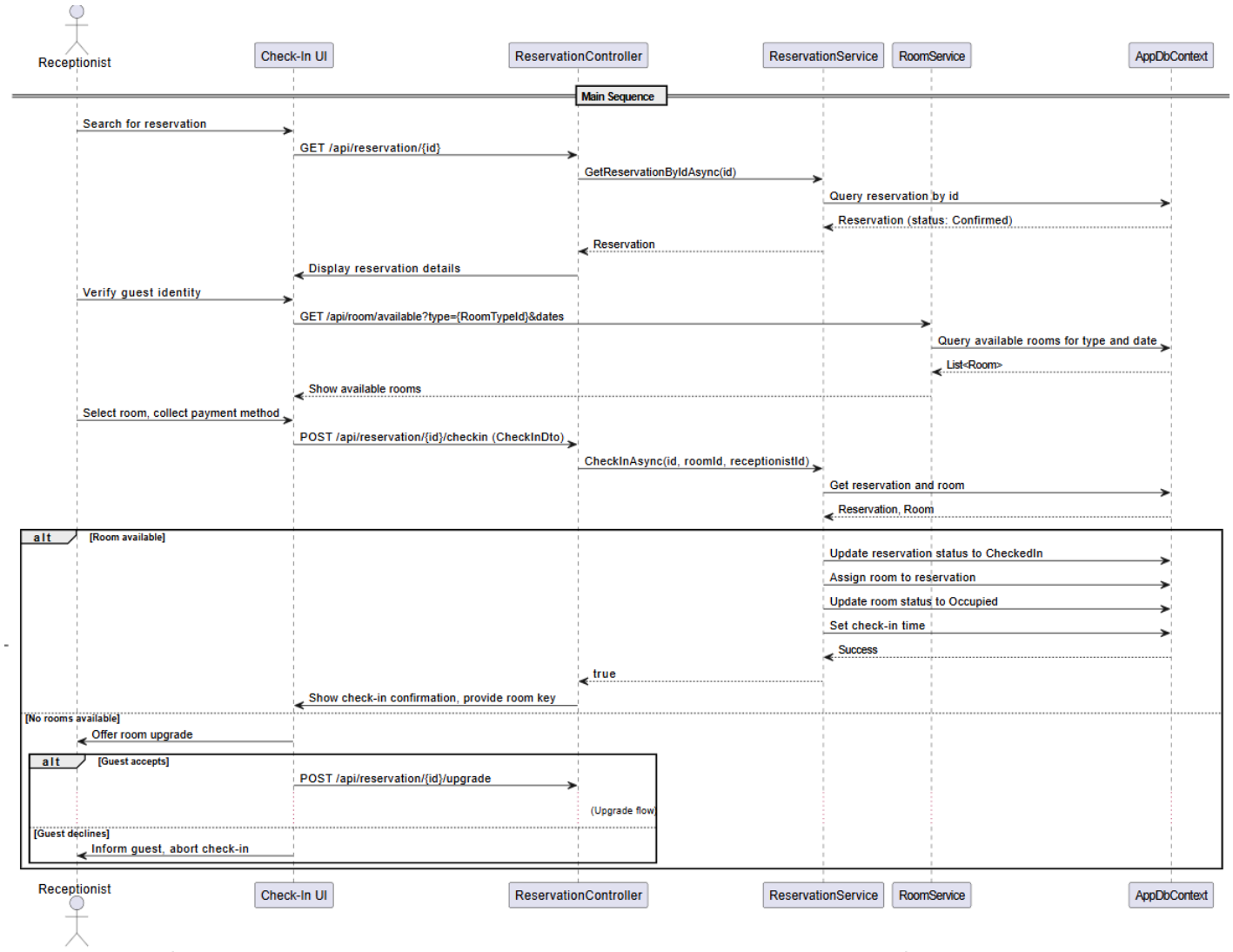
SeqD_05 - View Reservation



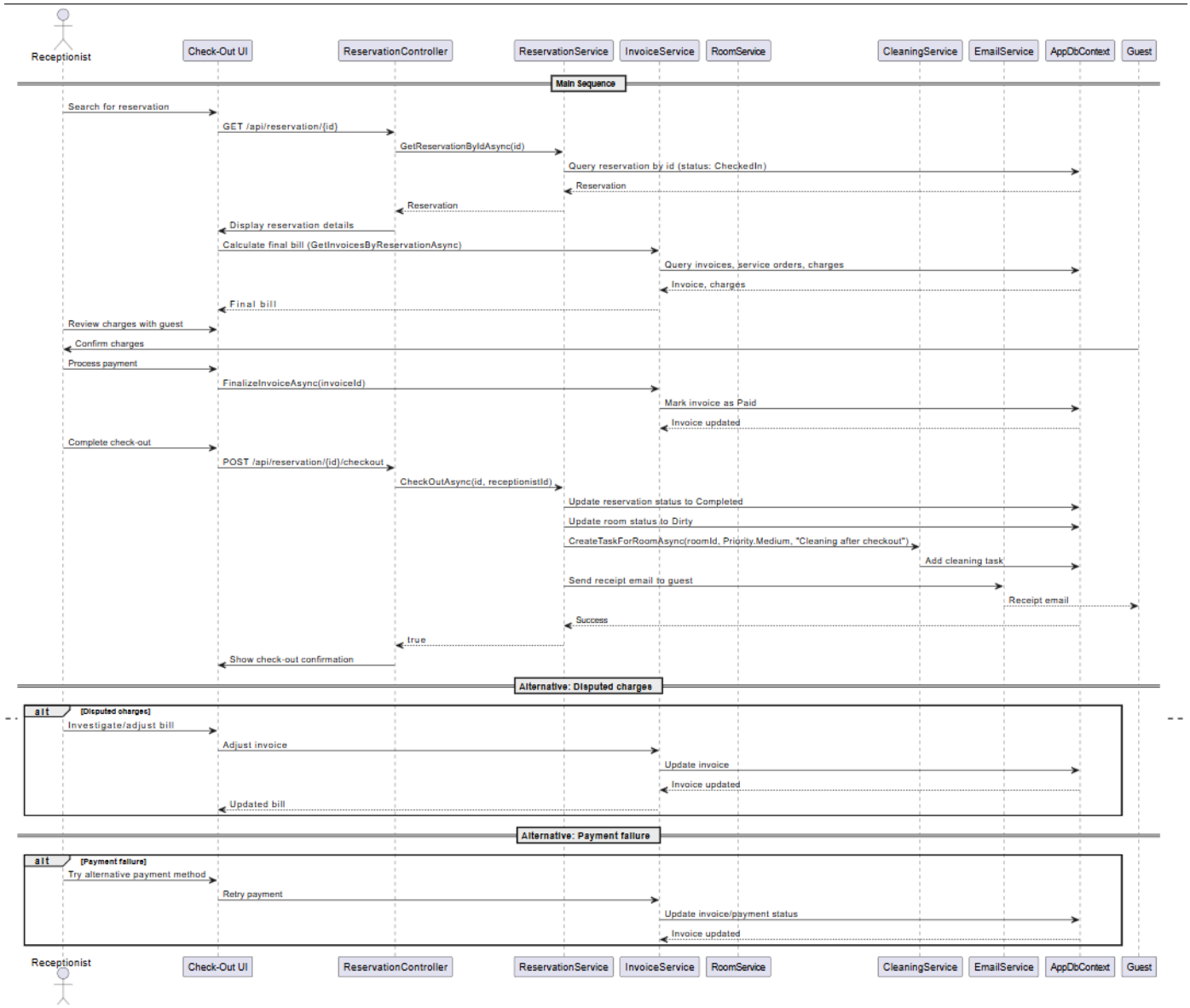
SeqD_06- Cancel reservation



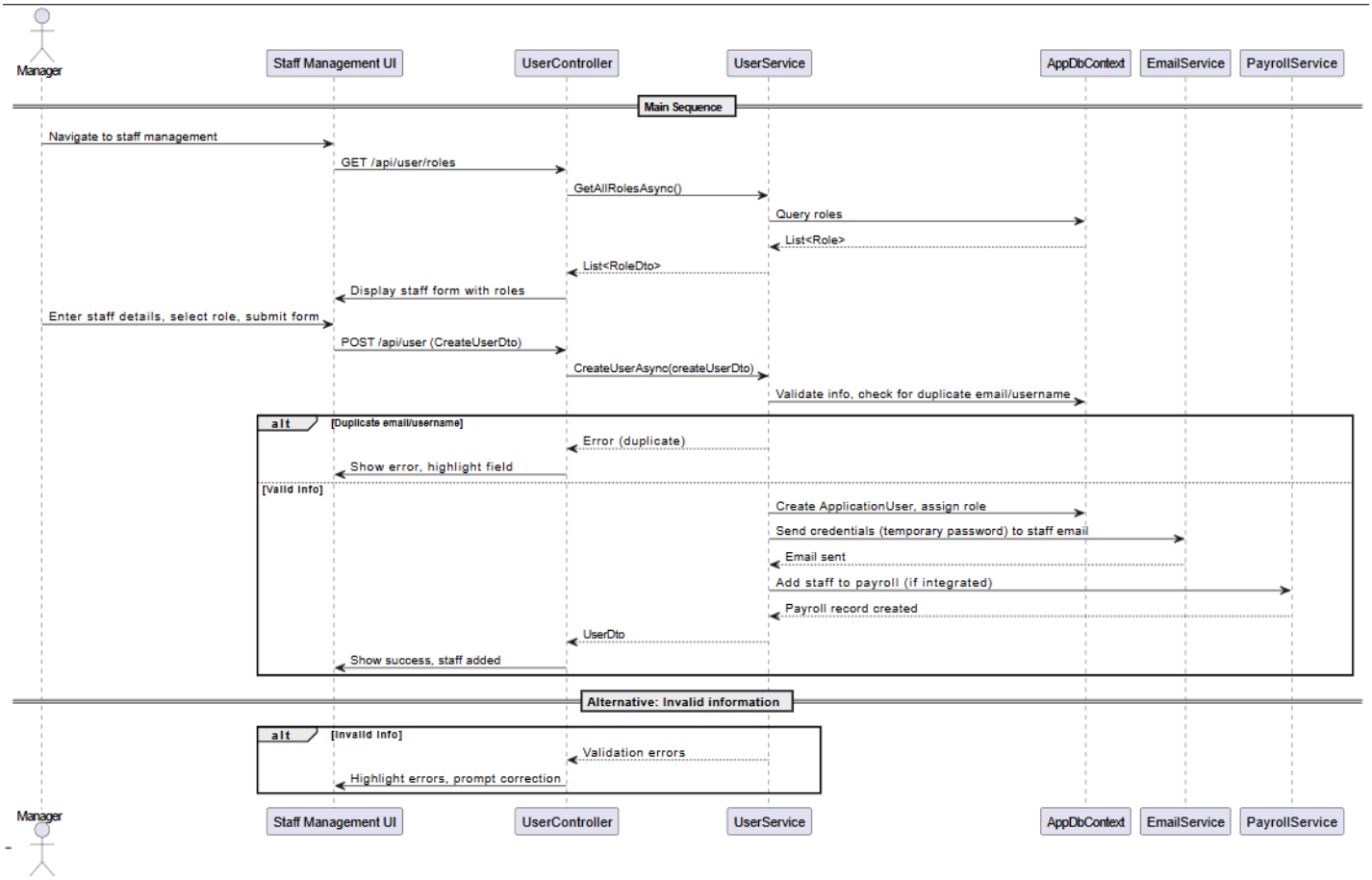
Seq_07 - Check-in-guest



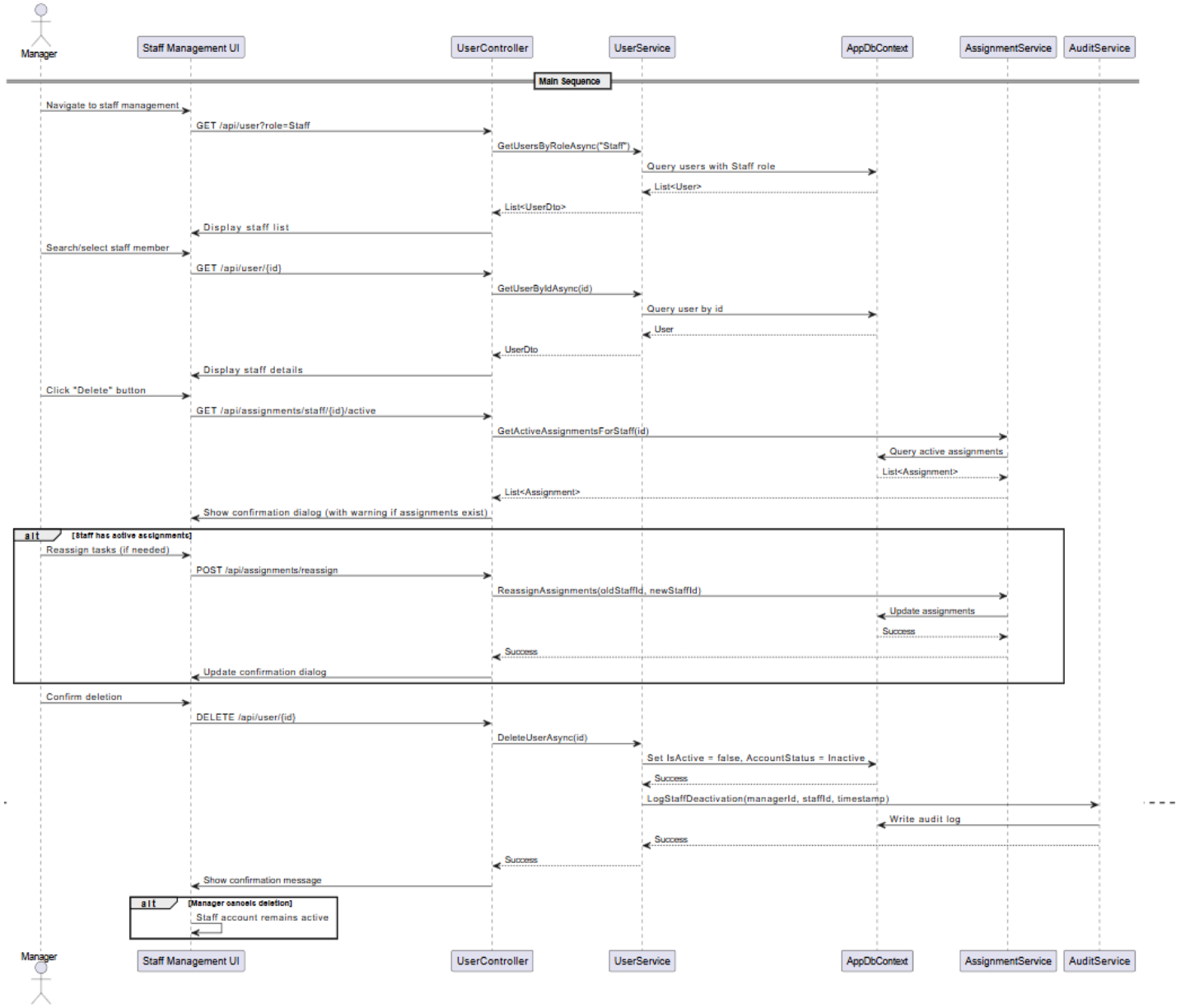
Seq_08 - Check-out-guest

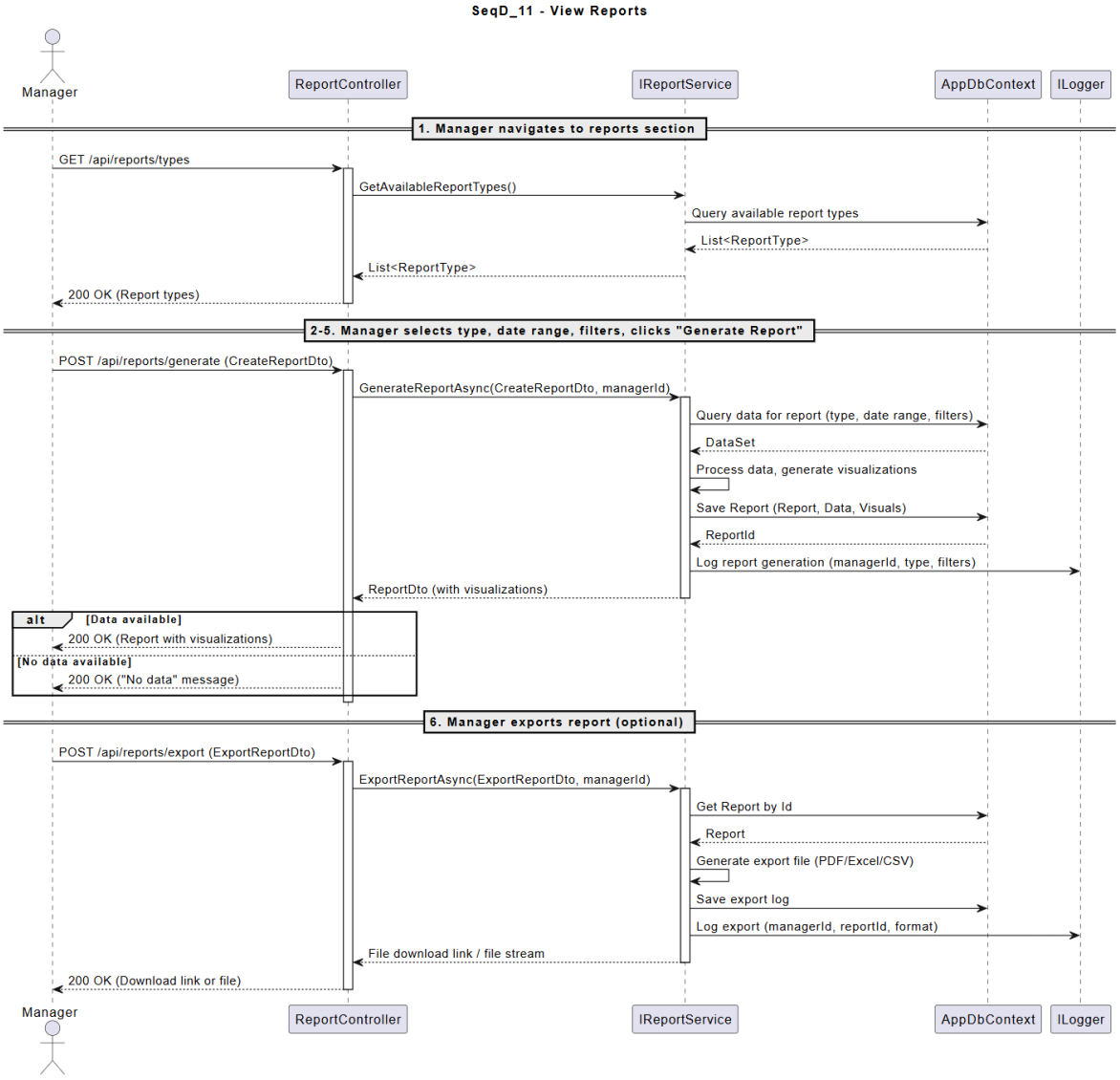


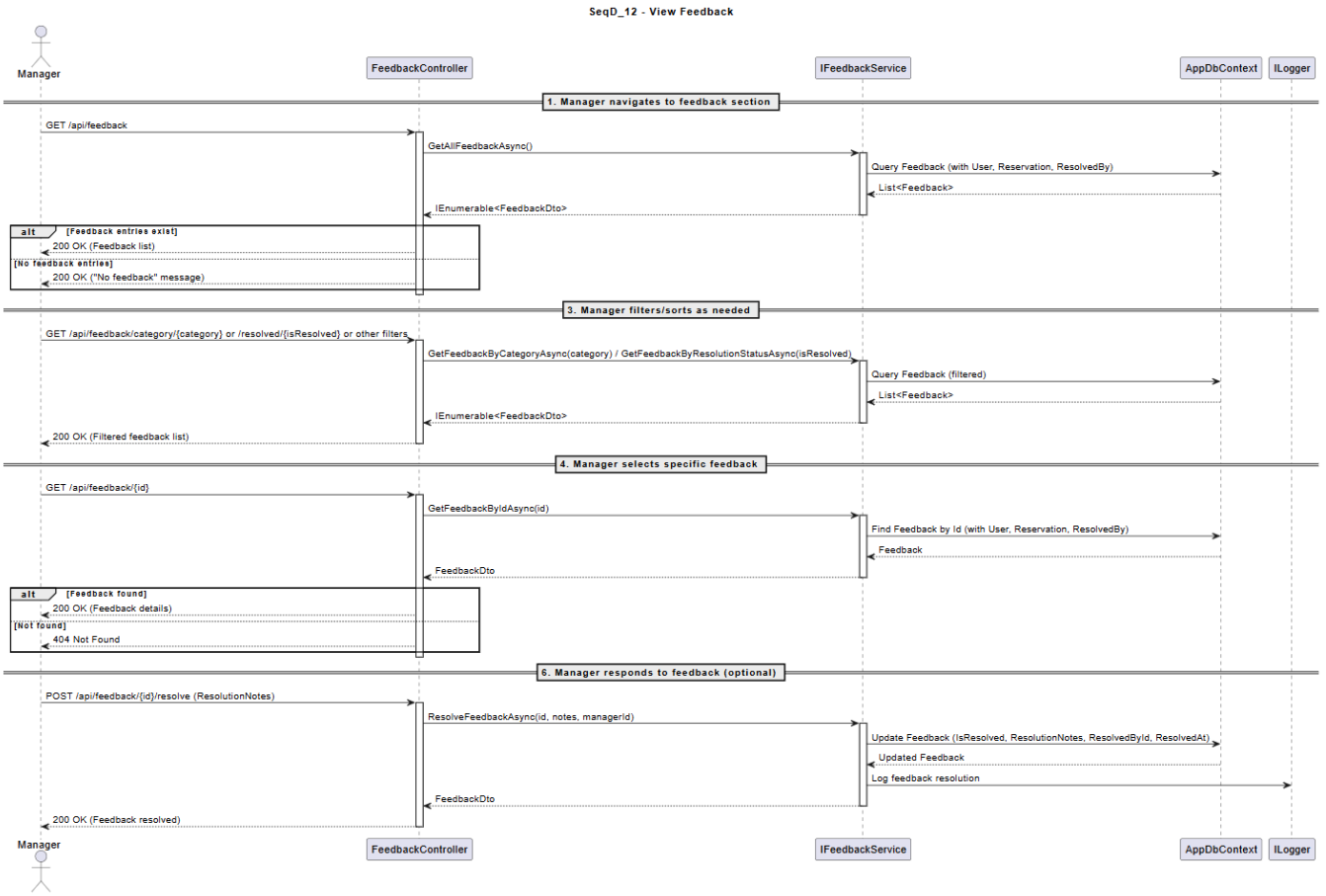
SeqD_9 - Add new staff member

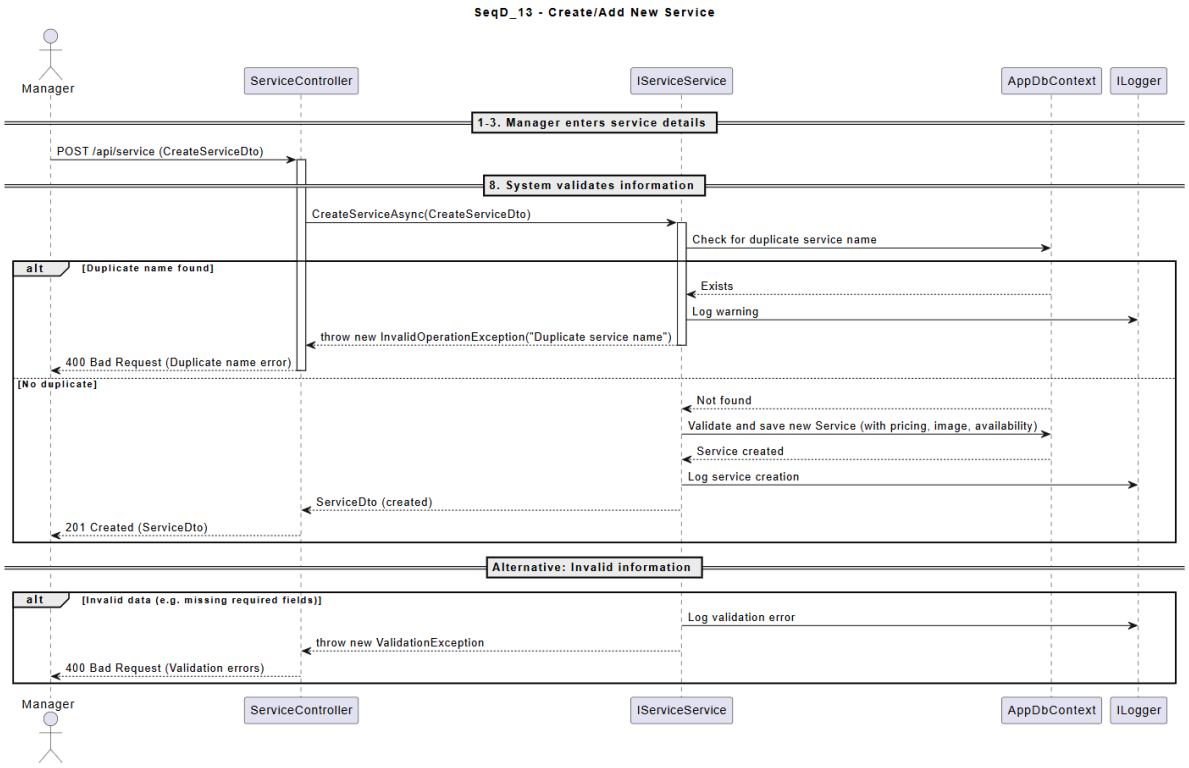


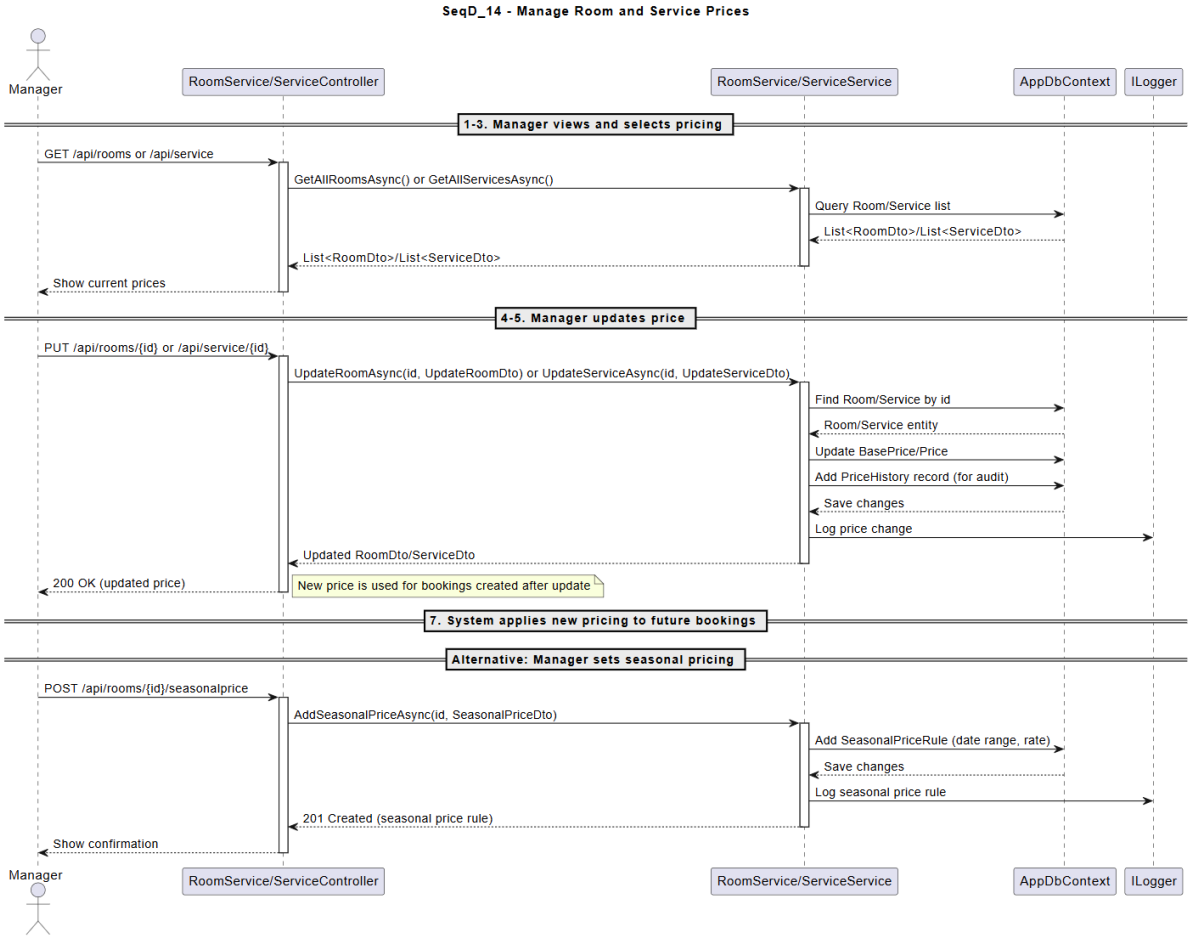
SeqD_10: Delete staff member

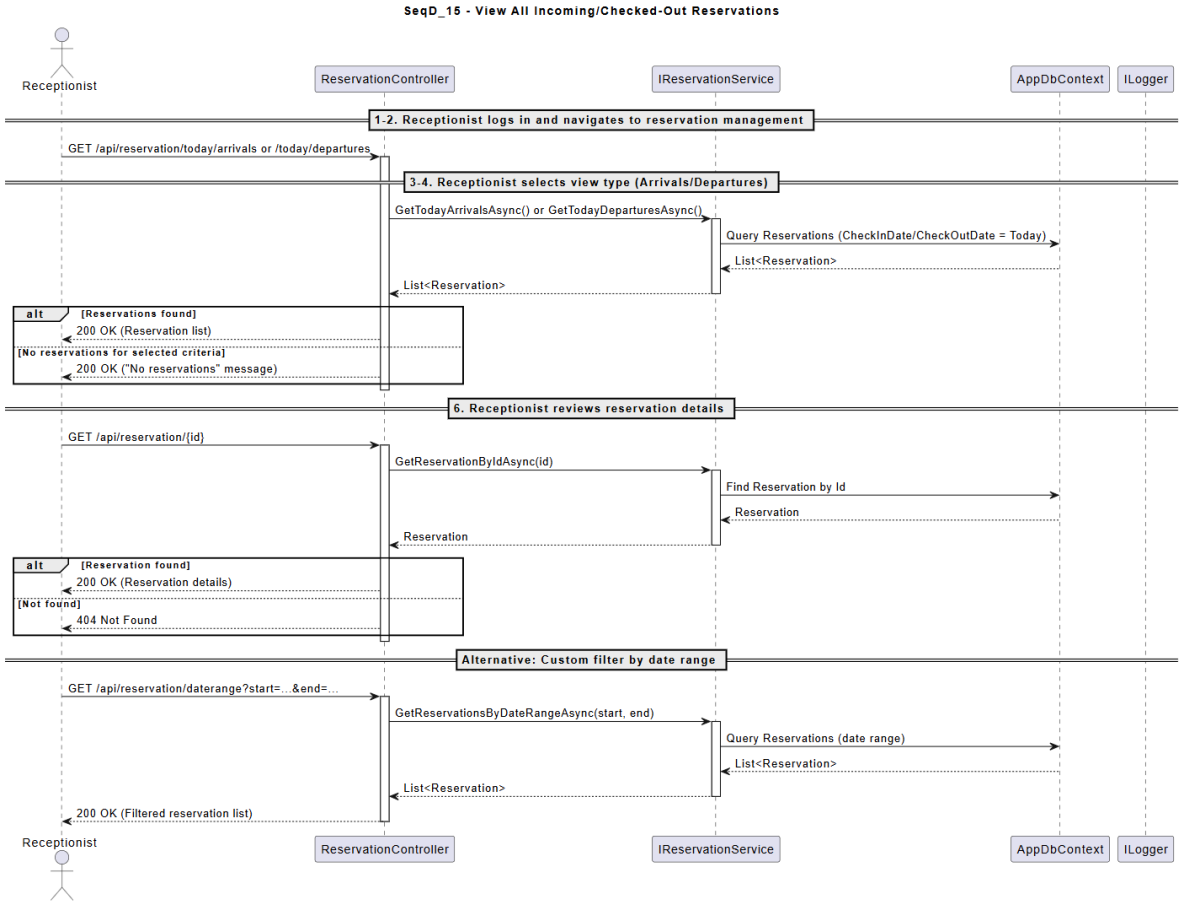


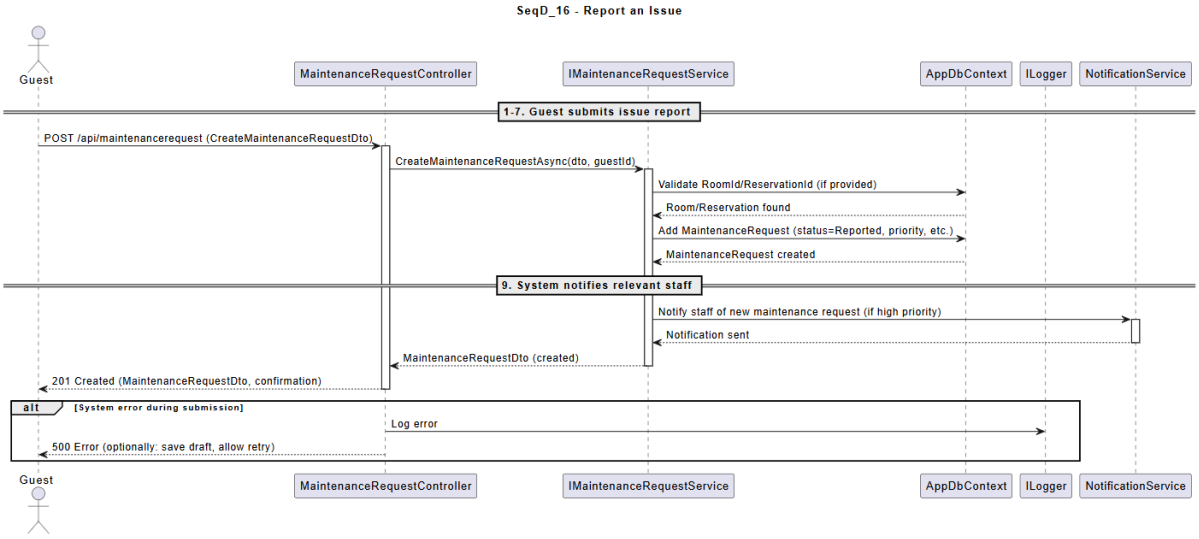

SeqD_11 - View Reports
Manager
ReportController
IReportService
AppDbContext
ILogger
1. Manager navigates to reports section
GET /api/reports/types
GetAvailableReportTypes()
Query available report types
List<ReportType>
List<ReportType>
200 OK (Report types)
2-5. Manager selects type, date range, filters, clicks "Generate Report"
POST /api/reports/generate (CreateReportDto)
GenerateReportAsync(CreateReportDto, managerId)
Query data for report (type, date range, filters)
DataSet
Process data, generate visualizations
Save Report (Report, Data, Visuals)
ReportId
Log report generation (managerId, type, filters)
ReportDto (with visualizations)
alt
[Data available]
200 OK (Report with visualizations)
[No data available]
200 OK ("No data" message)
6. Manager exports report (optional)
POST /api/reports/export (ExportReportDto)
ExportReportAsync(ExportReportDto, managerId)
Get Report by Id
Report
Generate export file (PDF/Excel/CSV)
Save export log
Log export (managerId, reportId, format)
File download link / file stream
200 OK (Download link or file)
Manager
ReportController
IReportService
AppDbContext
ILogger

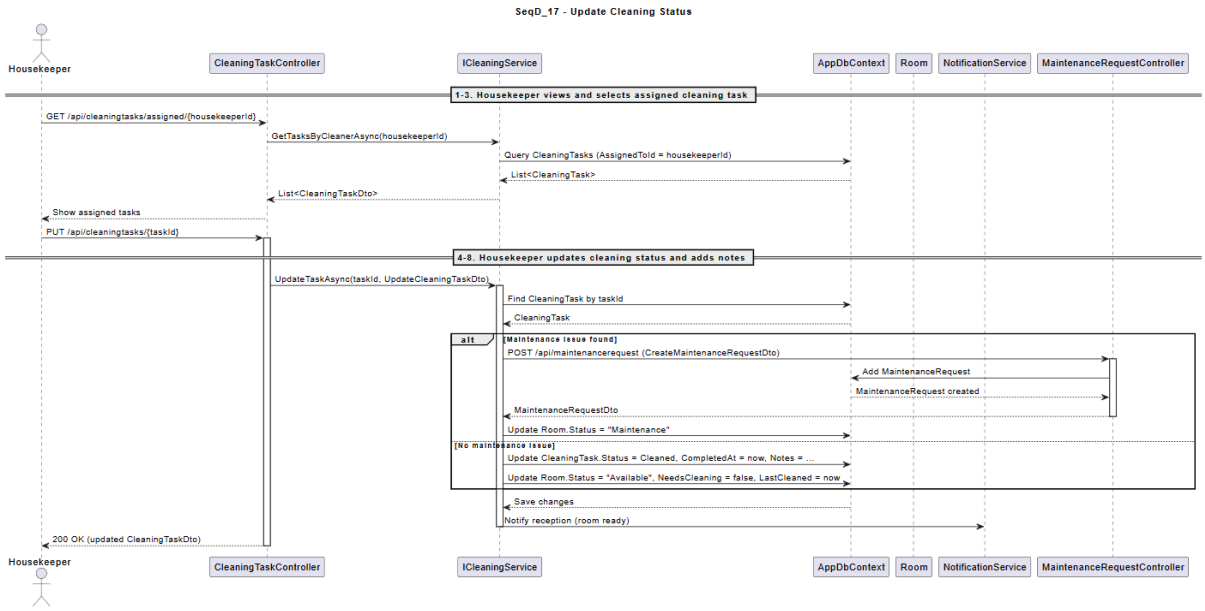

SeqD_12 - View Feedback
Manager
FeedbackController
IFeedbackService
AppDbContext
ILogger
1. Manager navigates to feedback section
GET /api/feedback
GetAllFeedbackAsync()
Query Feedback (with User, Reservation, ResolvedBy)
List<Feedback>
IEnumerable<FeedbackDto>
alt
[Feedback entries exist]
200 OK (Feedback list)
[No feedback entries]
200 OK ("No feedback" message)
3. Manager filters/sorts as needed
GET /api/feedback/category/{category} or /resolved/{isResolved} or other filters
GetFeedbackByCategoryAsync(category) / GetFeedbackByResolutionStatusAsync(isResolved)
Query Feedback (filtered)
List<Feedback>
IEnumerable<FeedbackDto>
200 OK (Filtered feedback list)
4. Manager selects specific feedback
GET /api/feedback/{id}
GetFeedbackByIdAsync(id)
Find Feedback by Id (with User, Reservation, ResolvedBy)
Feedback
FeedbackDto
alt
[Feedback found]
200 OK (Feedback details)
[Not found]
404 Not Found
6. Manager responds to feedback (optional)
POST /api/feedback/{id}/resolve (ResolutionNotes)
ResolveFeedbackAsync(id, notes, managerId)
Update Feedback (IsResolved, ResolutionNotes, ResolvedById, ResolvedAt)
Updated Feedback
Log feedback resolution
FeedbackDto
200 OK (Feedback resolved)
Manager
FeedbackController
IFeedbackService
AppDbContext
ILogger

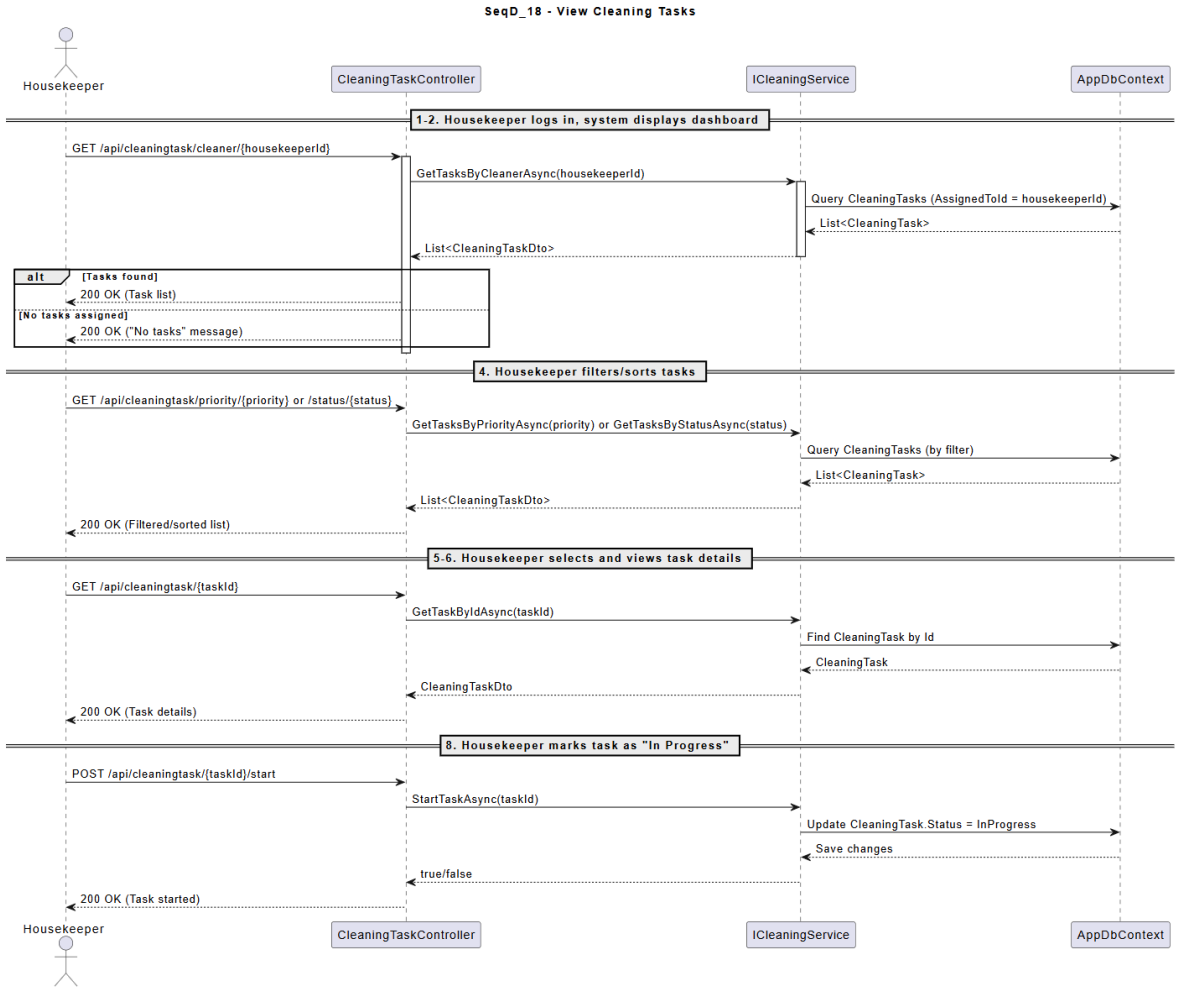

SeqD_13 - Create/Add New Service
Manager
ServiceController
IServiceService
AppDbContext
ILogger
1-3. Manager enters service details
POST /api/service (CreateServiceDto)
8. System validates information
CreateServiceAsync(CreateServiceDto)
Check for duplicate service name
alt
[Duplicate name found]
Exists
Log warning
throw new InvalidOperationException("Duplicate service name")
400 Bad Request (Duplicate name error)
[No duplicate]
Not found
Validate and save new Service (with pricing, image, availability)
Service created
Log service creation
ServiceDto (created)
201 Created (ServiceDto)
Alternative: Invalid information
alt
[Invalid data (e.g. missing required fields)]
Log validation error
throw new ValidationException
400 Bad Request (Validation errors)
Manager
ServiceController
IServiceService
AppDbContext
ILogger

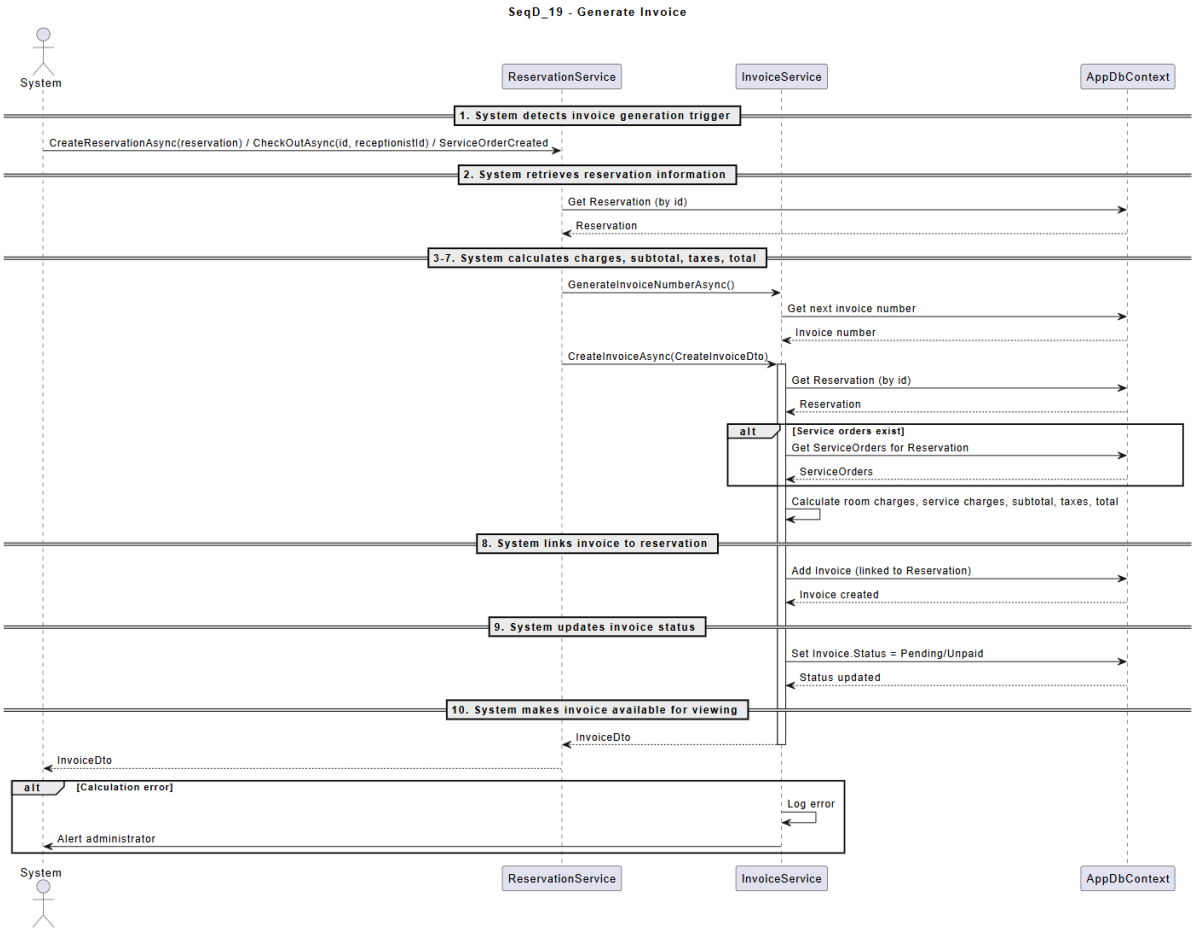

SeqD_14 - Manage Room and Service Prices
Manager
RoomService/ServiceController
RoomService/ServiceService
AppDbContext
ILogger
1-3. Manager views and selects pricing
GET /api/rooms or /api/service
GetAllRoomsAsync() or GetAllServicesAsync()
Query Room/Service list
List<RoomDto>/List<ServiceDto>
List<RoomDto>/List<ServiceDto>
Show current prices
4-5. Manager updates price
PUT /api/rooms/{id} or /api/service/{id}
UpdateRoomAsync(id, UpdateRoomDto) or UpdateServiceAsync(id, UpdateServiceDto)
Find Room/Service by id
Room/Service entity
Update BasePrice/Price
Add PriceHistory record (for audit)
Save changes
Log price change
Updated RoomDto/ServiceDto
200 OK (updated price)
New price is used for bookings created after update
7. System applies new pricing to future bookings
Alternative: Manager sets seasonal pricing
POST /api/rooms/{id}/seasonalprice
AddSeasonalPriceAsync(id, SeasonalPriceDto)
Add SeasonalPriceRule (date range, rate)
Save changes
Log seasonal price rule
201 Created (seasonal price rule)
Show confirmation
Manager
RoomService/ServiceController
RoomService/ServiceService
AppDbContext
ILogger

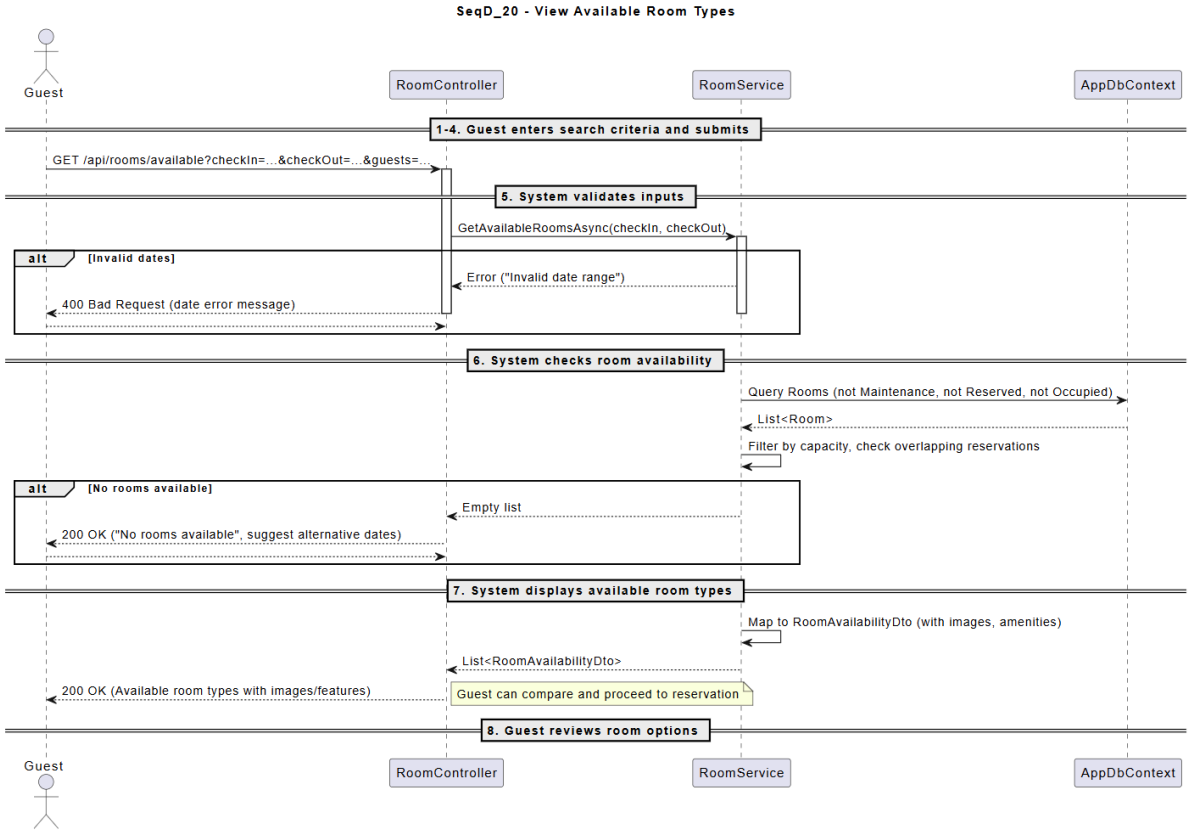

SeqD_15 - View All Incoming/Checked-Out Reservations
Receptionist
ReservationController
IReservationService
AppDbContext
ILogger
1-2. Receptionist logs in and navigates to reservation management
GET /api/reservation/today/arrivals or /today/departures
3-4. Receptionist selects view type (Arrivals/Departures)
GetTodayArrivalsAsync() or GetTodayDeparturesAsync()
Query Reservations (CheckInDate/CheckOutDate = Today)
List<Reservation>
List<Reservation>
alt
[Reservations found]
200 OK (Reservation list)
[No reservations for selected criteria]
200 OK ("No reservations" message)
6. Receptionist reviews reservation details
GET /api/reservation/{id}
GetReservationByIdAsync(id)
Find Reservation by Id
Reservation
Reservation
alt
[Reservation found]
200 OK (Reservation details)
[Not found]
404 Not Found
Alternative: Custom filter by date range
GET /api/reservation/daterange?start=...&end=...
GetReservationsByDateRangeAsync(start, end)
Query Reservations (date range)
List<Reservation>
List<Reservation>
200 OK (Filtered reservation list)
Receptionist
ReservationController
IReservationService
AppDbContext
ILogger


SeqD_16 - Report an Issue
Guest
MaintenanceRequestController
IMaintenanceRequestService
AppDbContext
ILogger
NotificationService
1-7. Guest submits issue report
POST /api/maintenancerequest (CreateMaintenanceRequestDto)
CreateMaintenanceRequestAsync(dto, guestId)
Validate RoomId/ReservationId (if provided)
Room/Reservation found
Add MaintenanceRequest (status=Reported, priority, etc.)
MaintenanceRequest created
9. System notifies relevant staff
Notify staff of new maintenance request (if high priority)
Notification sent
MaintenanceRequestDto (created)
201 Created (MaintenanceRequestDto, confirmation)
alt
[System error during submission]
Log error
500 Error (optionally: save draft, allow retry)
Guest
MaintenanceRequestController
IMaintenanceRequestService
AppDbContext
ILogger
NotificationService


SeqD_17 - Update Cleaning Status
Housekeeper
CleaningTaskController
ICleaningService
AppDbContext
Room
NotificationService
MaintenanceRequestController
1-3. Housekeeper views and selects assigned cleaning task
GET /api/cleaningtasks/assigned/{housekeeperId}
GetTasksByCleanerAsync(housekeeperId)
Query CleaningTasks (AssignedToId = housekeeperId)
List<CleaningTask>
List<CleaningTaskDto>
Show assigned tasks
PUT /api/cleaningtasks/{taskId}
4-8. Housekeeper updates cleaning status and adds notes
UpdateTaskAsync(taskId, UpdateCleaningTaskDto)
Find CleaningTask by taskId
CleaningTask
alt
[Maintenance issue found]
POST /api/maintenancerequest (CreateMaintenanceRequestDto)
Add MaintenanceRequest
MaintenanceRequest created
MaintenanceRequestDto
Update Room.Status = "Maintenance"
[No maintenance issue]
Update CleaningTask.Status = Cleaned, CompletedAt = now, Notes = ...
Update Room.Status = "Available", NeedsCleaning = false, LastCleaned = now
Save changes
Notify reception (room ready)
200 OK (updated CleaningTaskDto)

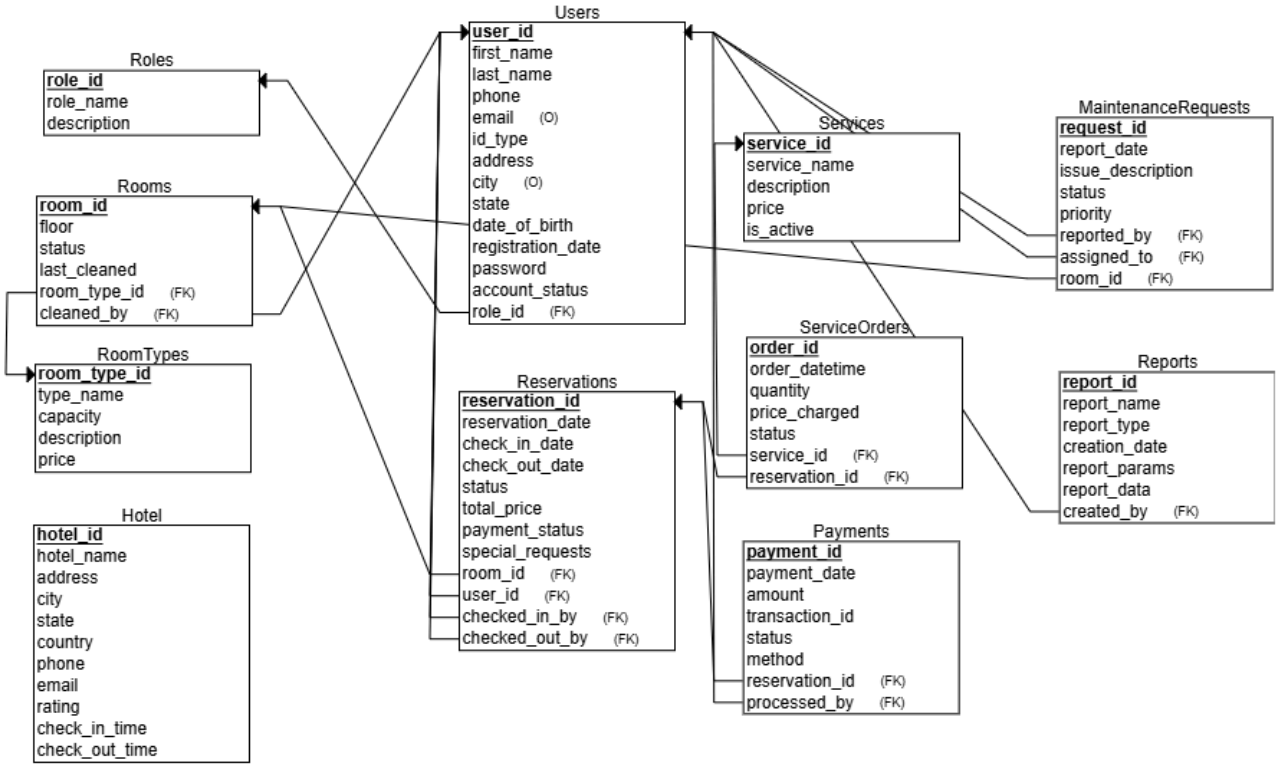

SeqD_18 - View Cleaning Tasks
Housekeeper
CleaningTaskController
ICleaningService
AppDbContext
1-2. Housekeeper logs in, system displays dashboard
GET /api/cleaningtask/cleaner/{housekeeperId}
GetTasksByCleanerAsync(housekeeperId)
Query CleaningTasks (AssignedToId = housekeeperId)
List<CleaningTask>
List<CleaningTaskDto>
alt
[Tasks found]
200 OK (Task list)
[No tasks assigned]
200 OK ("No tasks" message)
4. Housekeeper filters/sorts tasks
GET /api/cleaningtask/priority/{priority} or /status/{status}
GetTasksByPriorityAsync(priority) or GetTasksByStatusAsync(status)
Query CleaningTasks (by filter)
List<CleaningTask>
List<CleaningTaskDto>
200 OK (Filtered/sorted list)
5-6. Housekeeper selects and views task details
GET /api/cleaningtask/{taskId}
GetTaskByIdAsync(taskId)
Find CleaningTask by Id
CleaningTask
CleaningTaskDto
200 OK (Task details)
8. Housekeeper marks task as "In Progress"
POST /api/cleaningtask/{taskId}/start
StartTaskAsync(taskId)
Update CleaningTask.Status = InProgress
Save changes
true/false
200 OK (Task started)


SeqD_19 - Generate Invoice
System
ReservationService
InvoiceService
AppDbContext
1. System detects invoice generation trigger
CreateReservationAsync(reservation) / CheckOutAsync(id, receptionistId) / ServiceOrderCreated
2. System retrieves reservation information
Get Reservation (by id)
Reservation
3-7. System calculates charges, subtotal, taxes, total
GenerateInvoiceNumberAsync()
Get next invoice number
Invoice number
CreateInvoiceAsync(CreateInvoiceDto)
Get Reservation (by id)
Reservation
alt
[Service orders exist]
Get ServiceOrders for Reservation
ServiceOrders
Calculate room charges, service charges, subtotal, taxes, total
8. System links invoice to reservation
Add Invoice (linked to Reservation)
Invoice created
9. System updates invoice status
Set Invoice.Status = Pending/Unpaid
Status updated
10. System makes invoice available for viewing
InvoiceDto
InvoiceDto
alt
[Calculation error]
Log error
Alert administrator
System
ReservationService
InvoiceService
AppDbContext

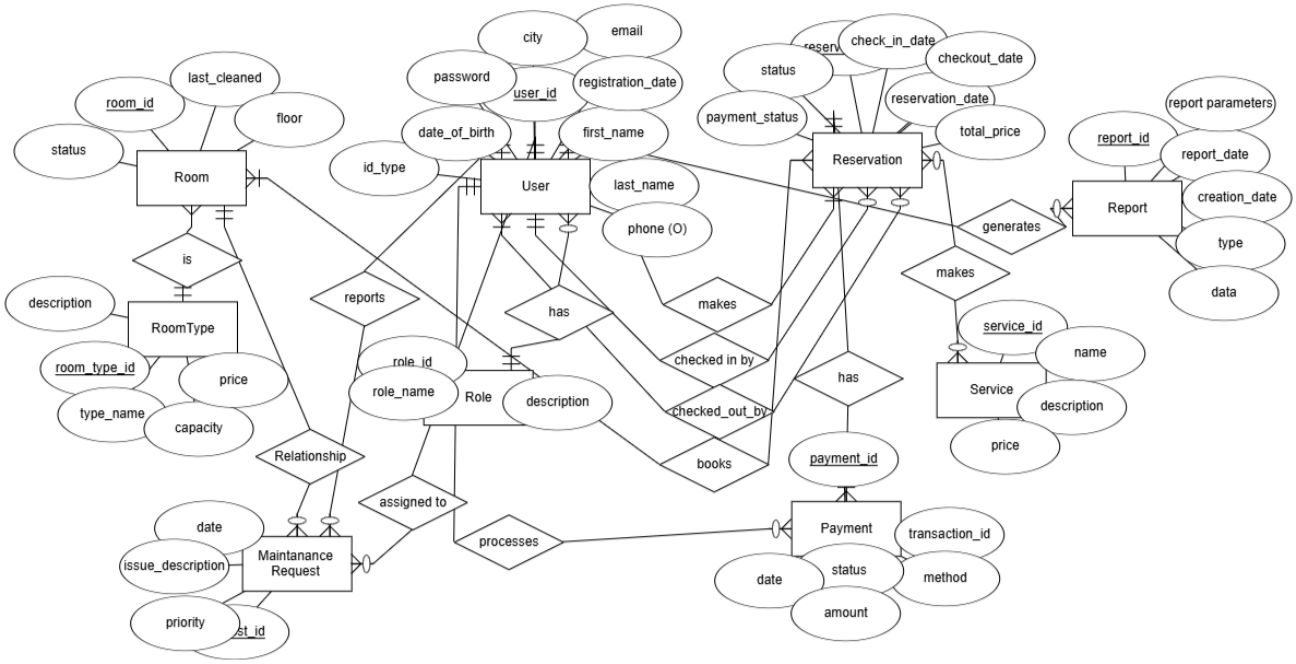

SeqD_20 - View Available Room Types
Guest
RoomController
RoomService
AppDbContext
1-4. Guest enters search criteria and submits
GET /api/rooms/available?checkIn=...&checkOut=...&guests=...
5. System validates inputs
GetAvailableRoomsAsync(checkIn, checkOut)
alt
[Invalid dates]
Error ("Invalid date range")
400 Bad Request (date error message)
6. System checks room availability
Query Rooms (not Maintenance, not Reserved, not Occupied)
List<Room>
Filter by capacity, check overlapping reservations
alt
[No rooms available]
Empty list
200 OK ("No rooms available", suggest alternative dates)
7. System displays available room types
Map to RoomAvailabilityDto (with images, amenities)
List<RoomAvailabilityDto>
200 OK (Available room types with images/features)
Guest can compare and proceed to reservation
8. Guest reviews room options
Guest
RoomController
RoomService
AppDbContext

4.4 Entity Relation

4.4.1 Database Schema Design



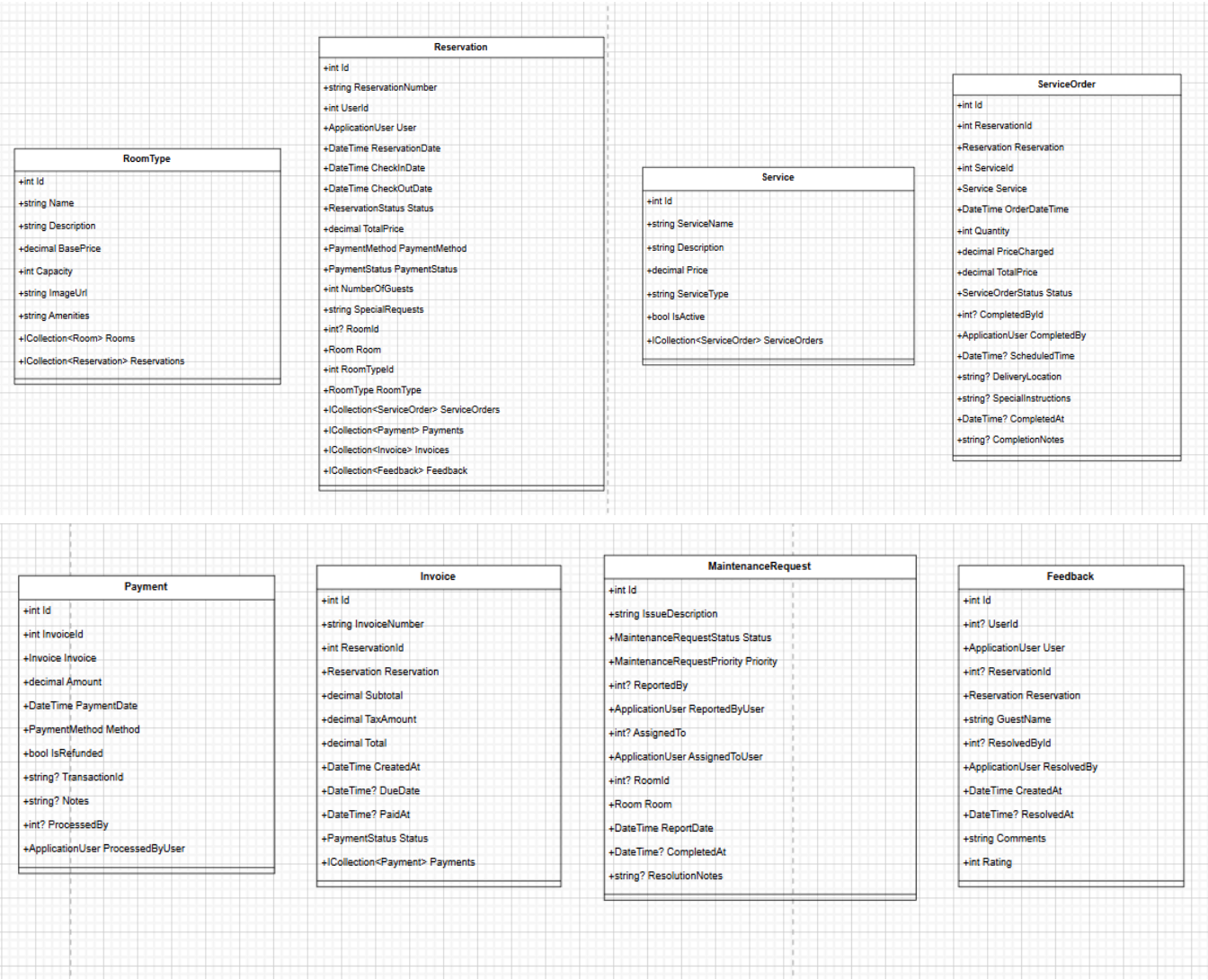
4.4.2 Entity Relation Diagram

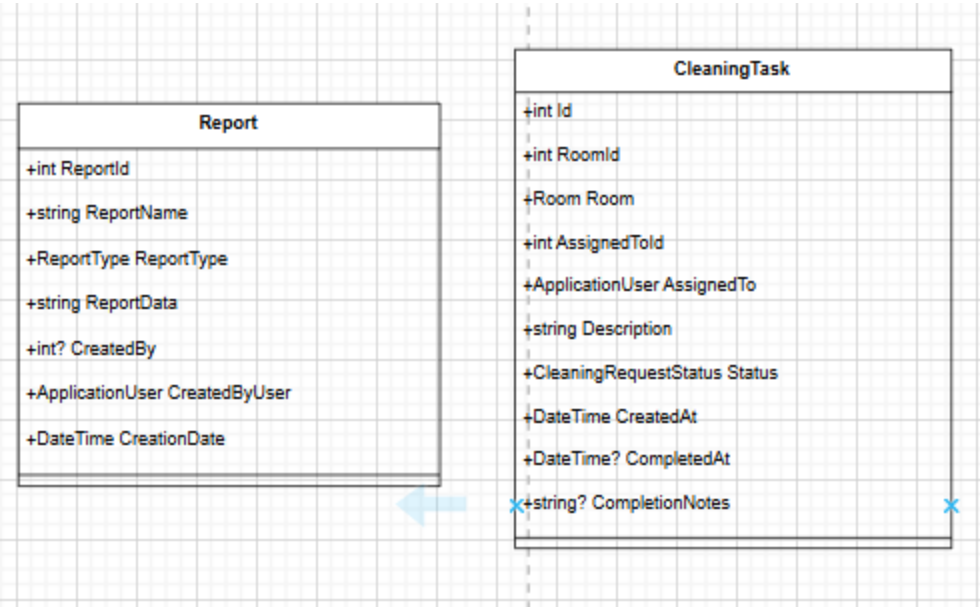


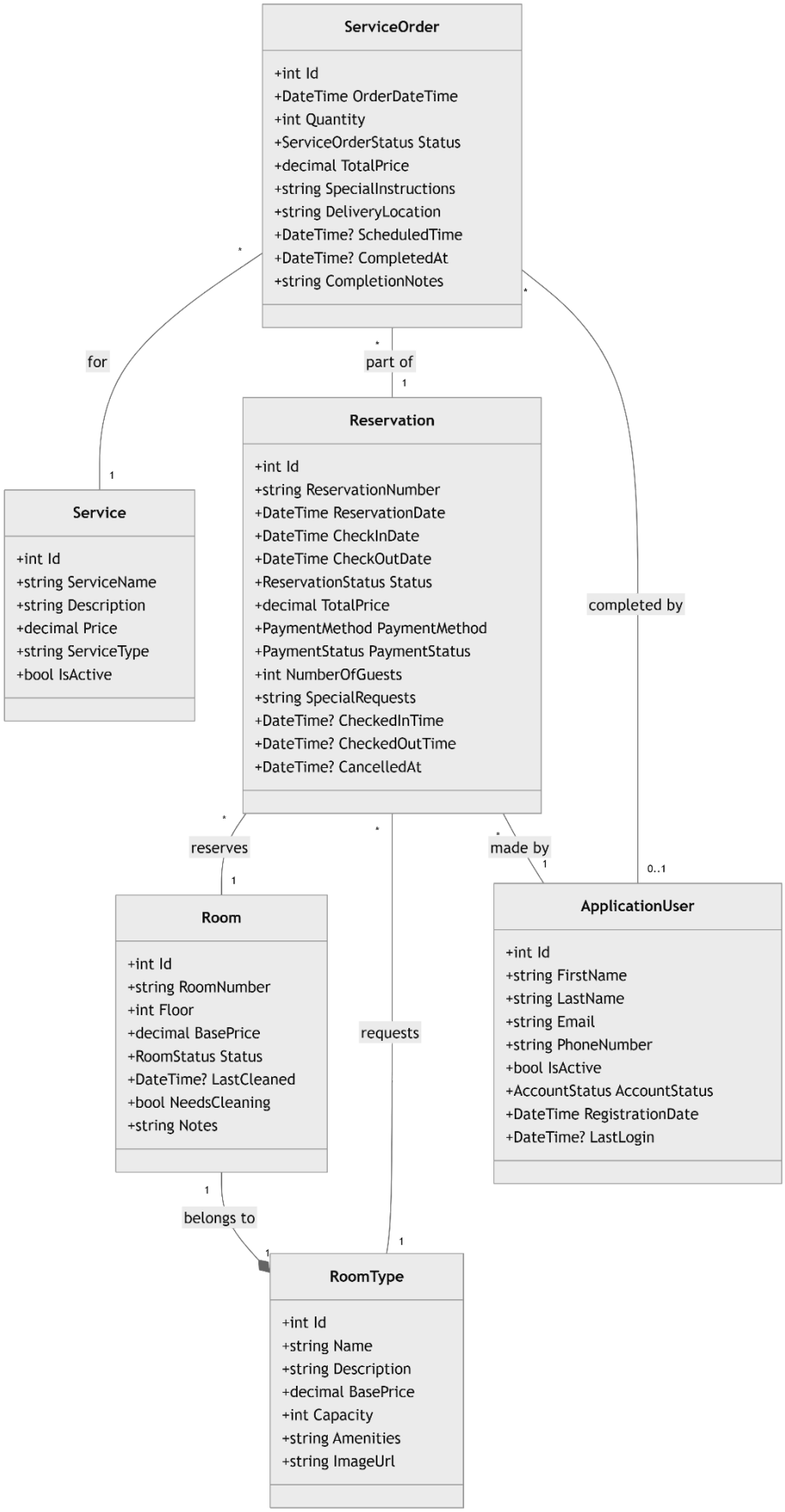
4.5 Structural Diagrams

4.5.1 Class Diagram

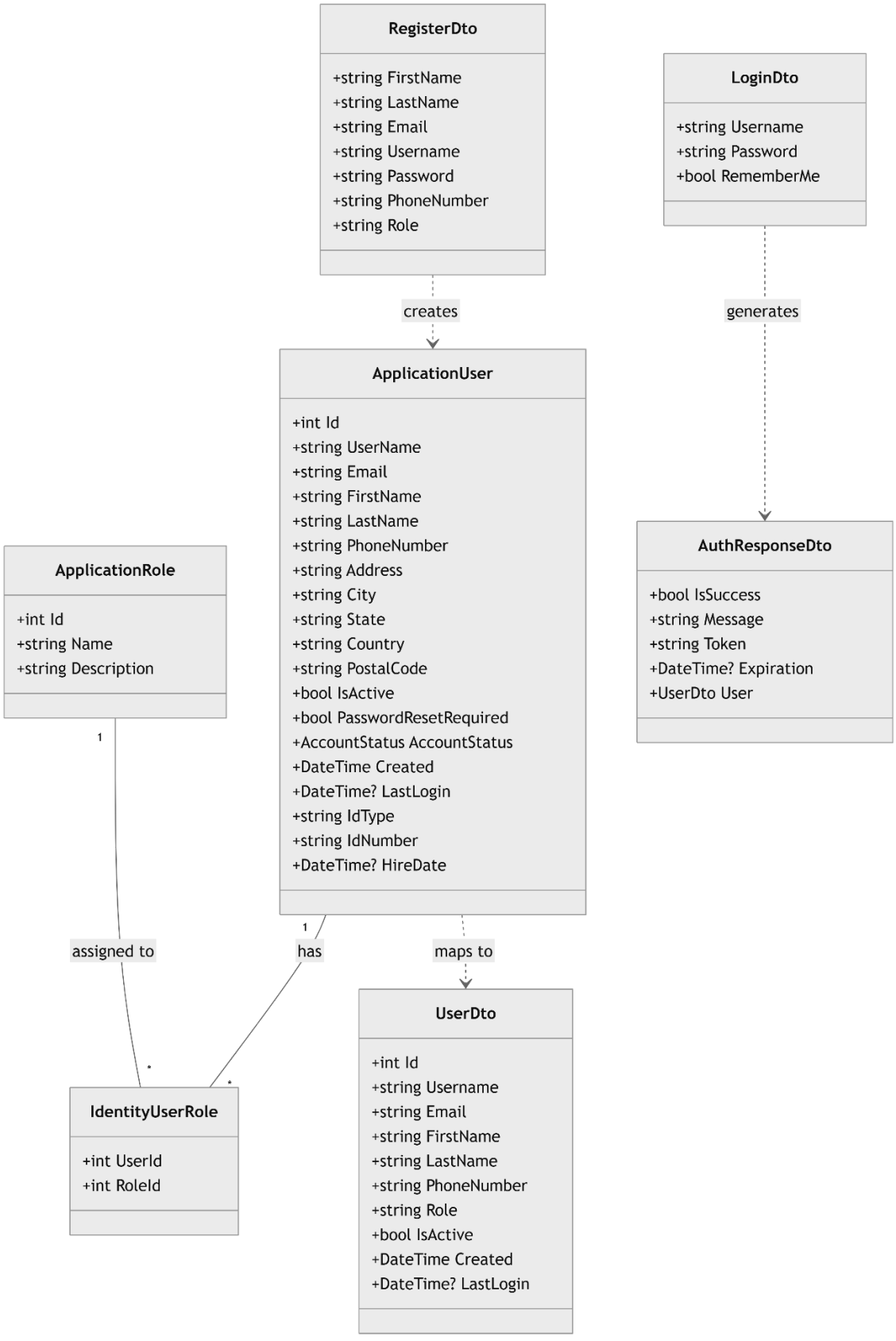
c



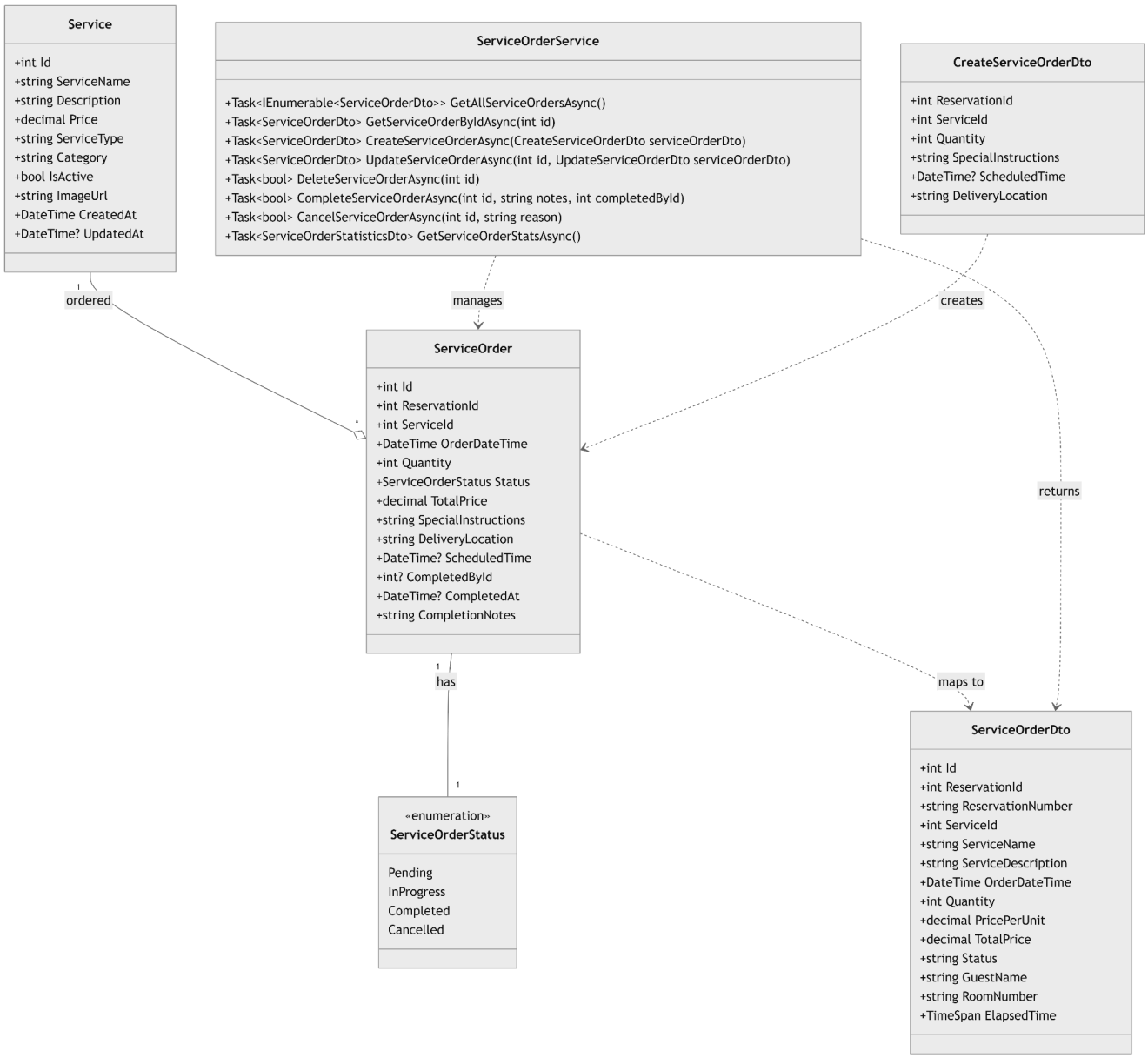

Report
+int ReportId
+string ReportName
+ReportType ReportType
+string ReportData
+int? CreatedBy
+ApplicationUser CreatedByUser
+DateTime CreationDate
CleaningTask
+int Id
+int RoomId
+Room Room
+int AssignedToId
+ApplicationUser AssignedTo
+string Description
+CleaningRequestStatus Status
+DateTime CreatedAt
+DateTime? CompletedAt
+string? CompletionNotes


ServiceOrder
+int Id
+DateTime OrderDateTime
+int Quantity
+ServiceOrderStatus Status
+decimal TotalPrice
+string SpecialInstructions
+string DeliveryLocation
+DateTime? ScheduledTime
+DateTime? CompletedAt
+string CompletionNotes
for
part of
completed by
Service
+int Id
+string ServiceName
+string Description
+decimal Price
+string ServiceType
+bool IsActive
Reservation
+int Id
+string ReservationNumber
+DateTime ReservationDate
+DateTime CheckInDate
+DateTime CheckOutDate
+ReservationStatus Status
+decimal TotalPrice
+PaymentMethod PaymentMethod
+PaymentStatus PaymentStatus
+int NumberOfGuests
+string SpecialRequests
+DateTime? CheckedInTime
+DateTime? CheckedOutTime
+DateTime? CancelledAt
reserves
made by
requests
Room
+int Id
+string RoomNumber
+int Floor
+decimal BasePrice
+RoomStatus Status
+DateTime? LastCleaned
+bool NeedsCleaning
+string Notes
ApplicationUser
+int Id
+string FirstName
+string LastName
+string Email
+string PhoneNumber
+bool IsActive
+AccountStatus AccountStatus
+DateTime RegistrationDate
+DateTime? LastLogin
belongs to
RoomType
+int Id
+string Name
+string Description
+decimal BasePrice
+int Capacity
+string Amenities
+string ImageUrl

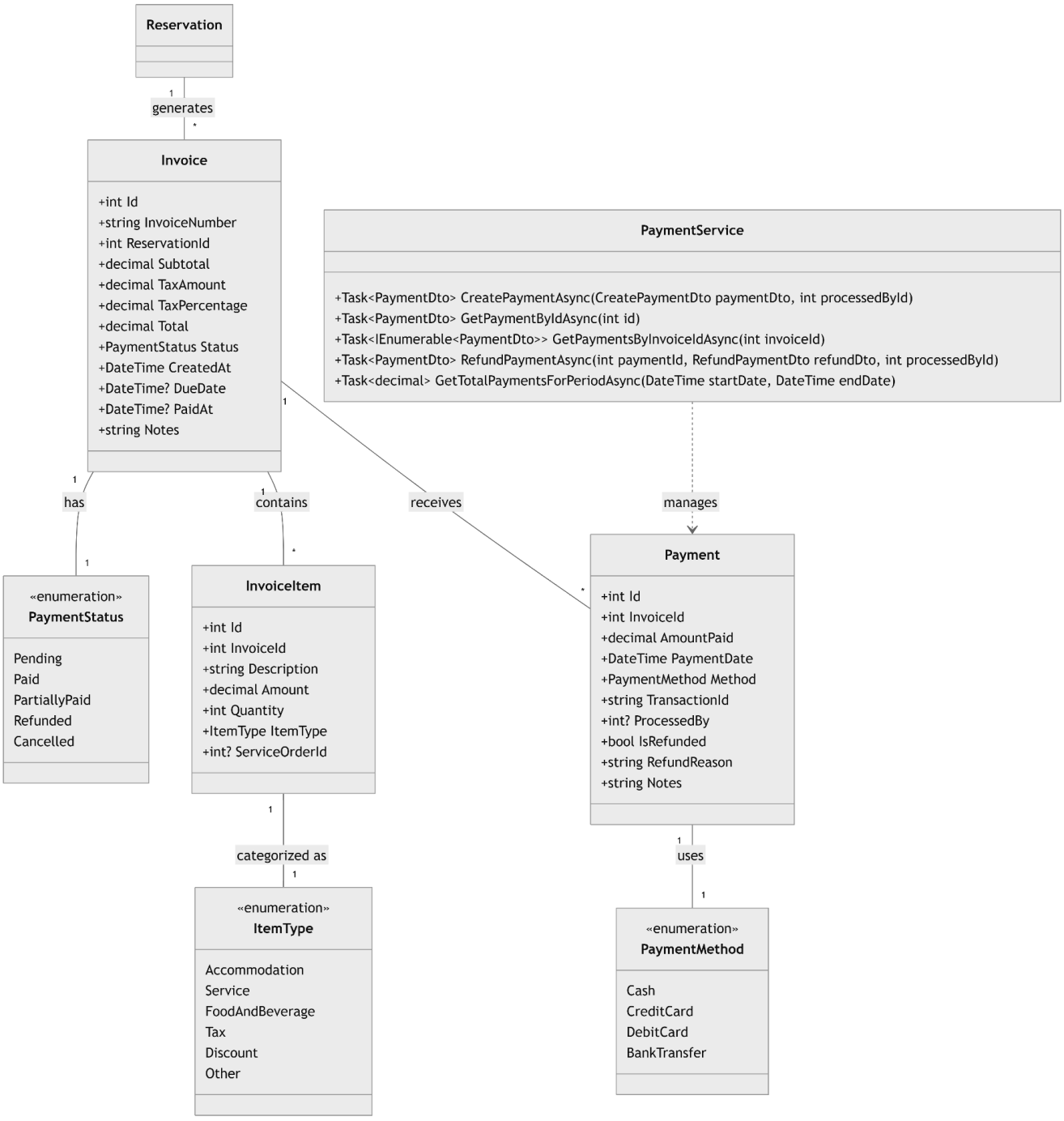
User Management Classes



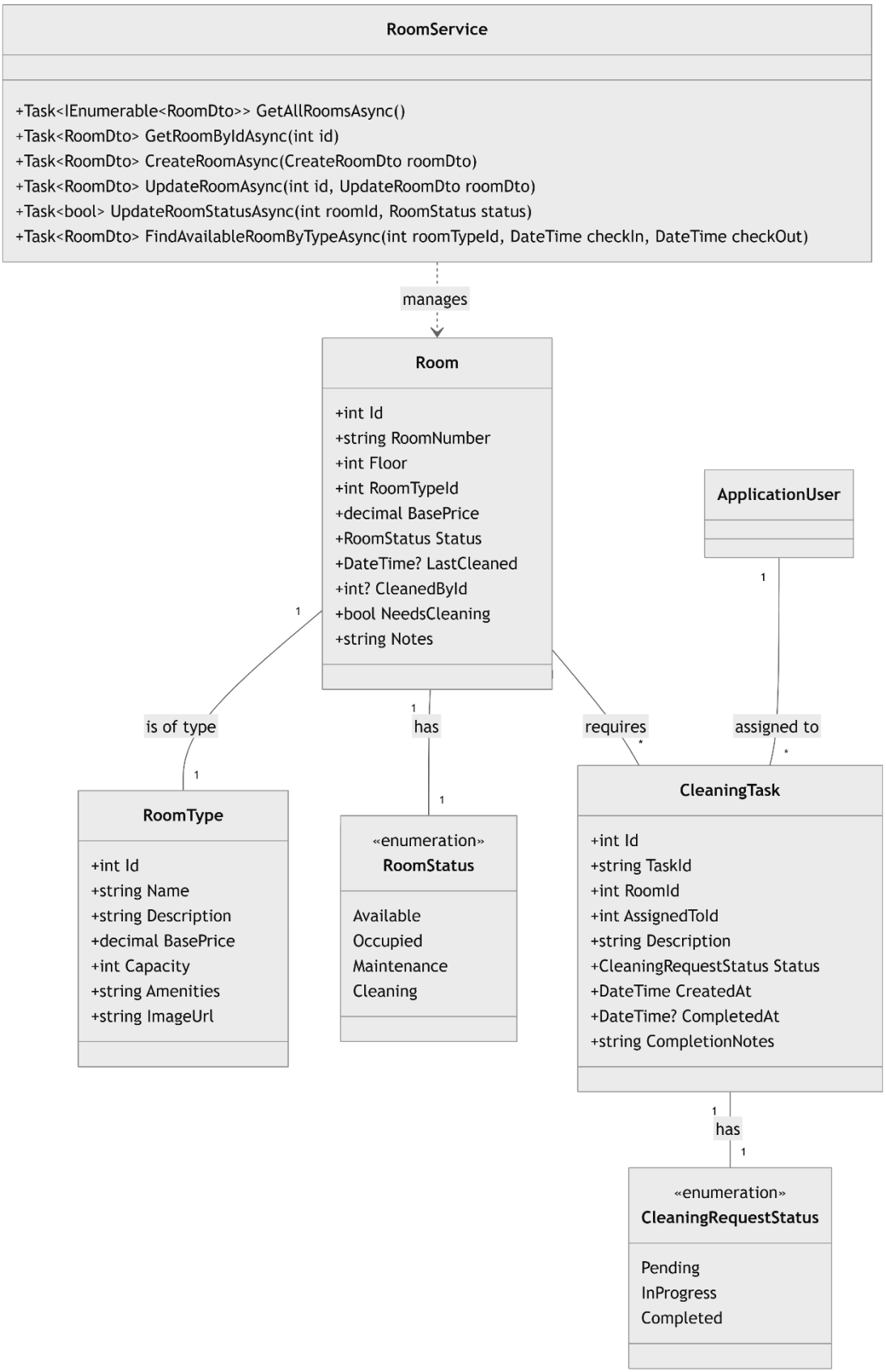
Service Order Management - Class Diagram



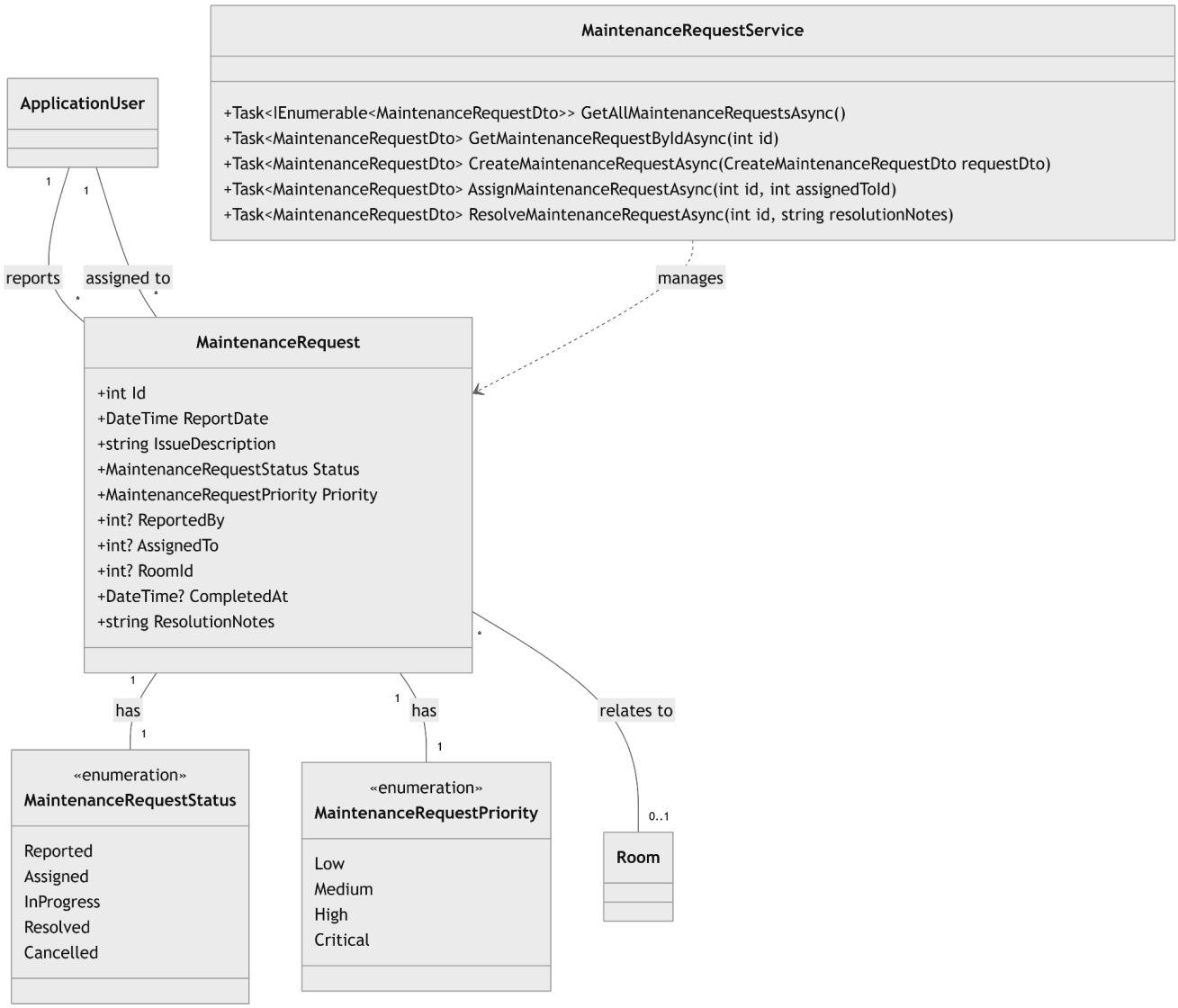
Invoice and Payment Management - Class Diagram



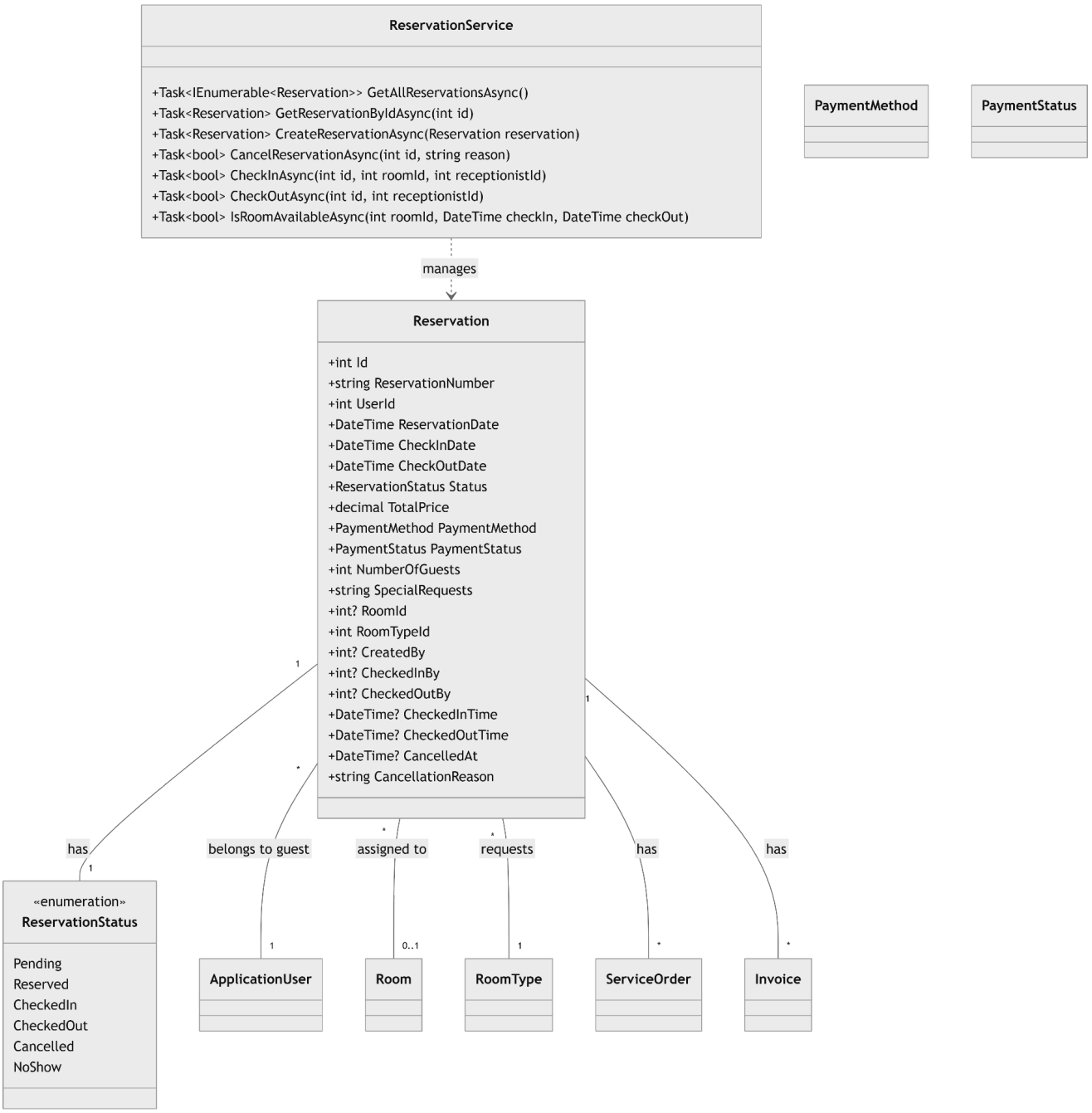
Room and Cleaning Management - Class Diagram

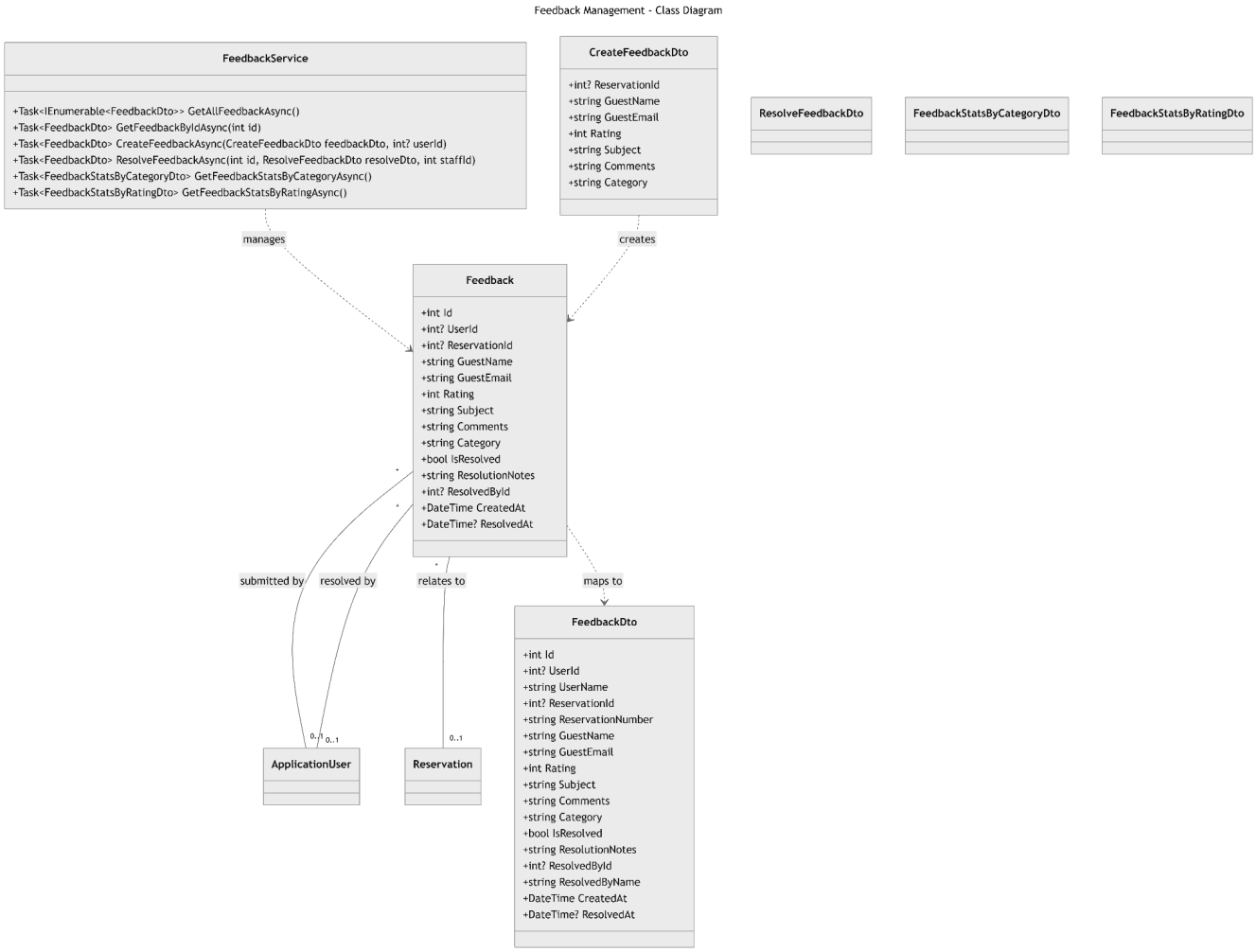


Maintenance Request Management - Class Diagram

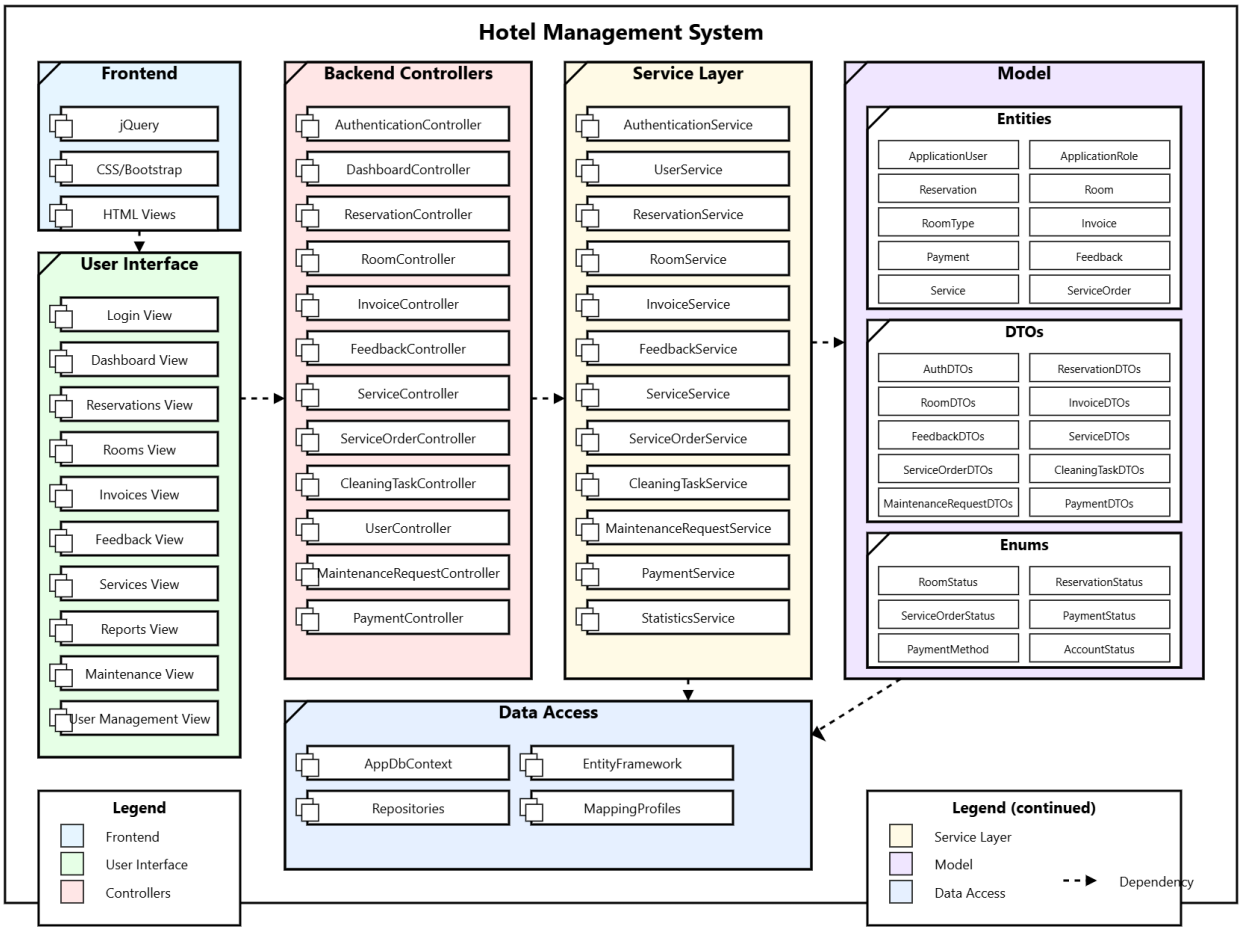


Reservation Management - Class Diagram

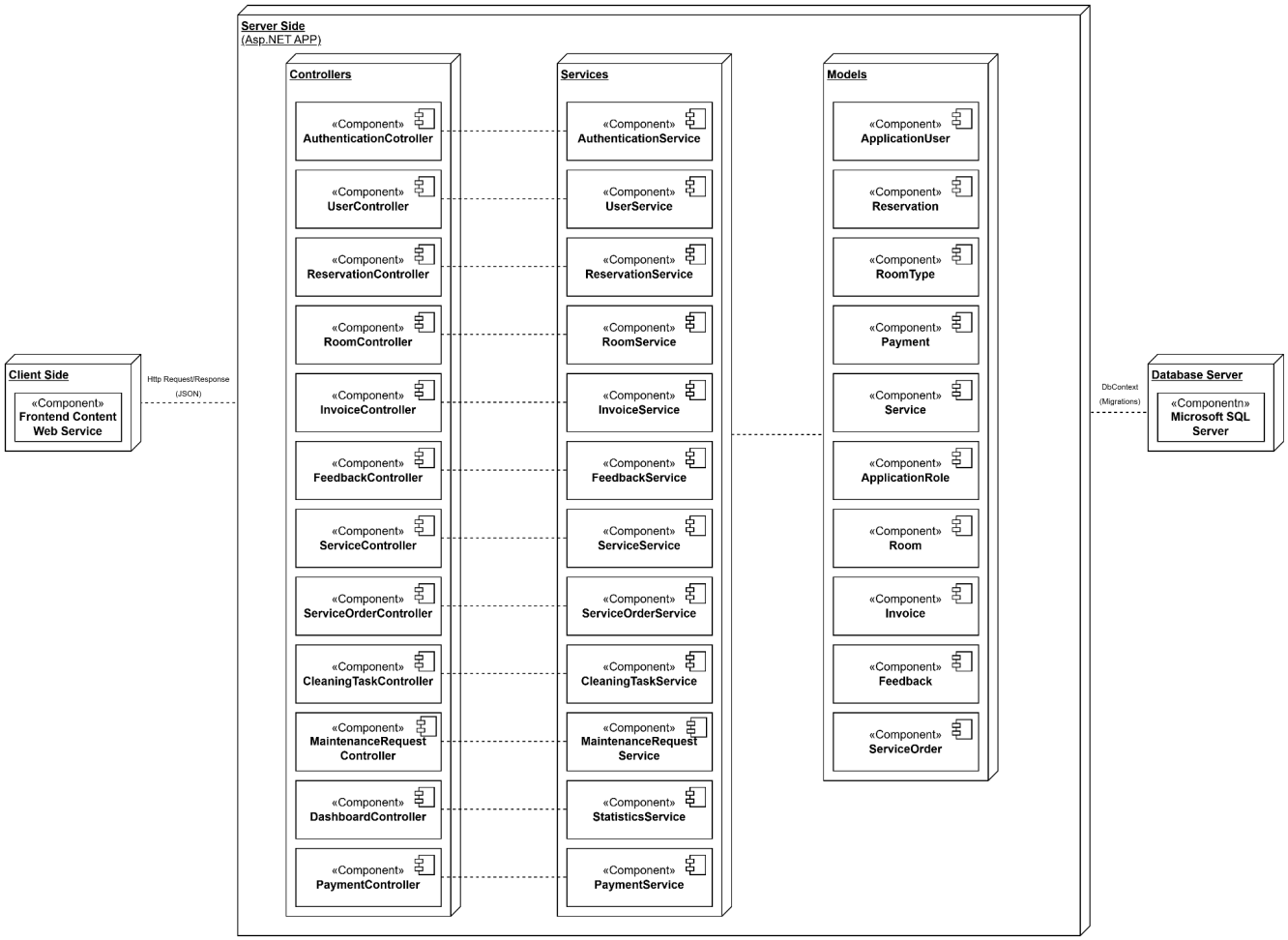



Feedback Management - Class Diagram
FeedbackService
+Task<IEnumerable<FeedbackDto>> GetAllFeedbackAsync()
+Task<FeedbackDto> GetFeedbackByIdAsync(int id)
+Task<FeedbackDto> CreateFeedbackAsync(CreateFeedbackDto feedbackDto, int? userId)
+Task<FeedbackDto> ResolveFeedbackAsync(int id, ResolveFeedbackDto resolveDto, int staffId)
+Task<FeedbackStatsByCategoryDto> GetFeedbackStatsByCategoryAsync()
+Task<FeedbackStatsByRatingDto> GetFeedbackStatsByRatingAsync()
CreateFeedbackDto
+int? ReservationId
+string GuestName
+string GuestEmail
+int Rating
+string Subject
+string Comments
+string Category
ResolveFeedbackDto
FeedbackStatsByCategoryDto
FeedbackStatsByRatingDto
manages
creates
Feedback
+Int Id
+int? UserId
+int? ReservationId
+string GuestName
+string GuestEmail
+int Rating
+string Subject
+string Comments
+string Category
+bool IsResolved
+string ResolutionNotes
+int? ResolvedById
+DateTime CreatedAt
+DateTime? ResolvedAt
submitted by
resolved by
relates to
maps to
0..1
0..1
0..1
ApplicationUser
Reservation
FeedbackDto
+int Id
+int? UserId
+string UserName
+int? ReservationId
+string ReservationNumber
+string GuestName
+string GuestEmail
+int Rating
+string Subject
+string Comments
+string Category
+bool IsResolved
+string ResolutionNotes
+int? ResolvedById
+string ResolvedByName
+DateTime CreatedAt
+DateTime? ResolvedAt

4.5.3 Component Diagrams



4.5.4 Deployment Diagram



5. Implementation Technology

Implementation Technology

HMS (Hotel Management System) is a dynamic Web Application. For the creation of this software, we have combined **Client Side Scripting** and **Server Side Scripting**. The communication between the client and the server takes place via **HTTPS protocol**.

For the **Client Side**, we have used the following technologies:

- JavaScript
- TypeScript
- jQuery (JavaScript Framework Library)

For the **Server Side**, we have utilized **ASP.NET 8**, which enables seamless interactions within the software and amongst users. Embracing an **Object-Oriented Programming (OOP)** approach, our architecture incorporates a robust set of features including **models, migrations, controllers, services, and cryptography**. These components ensure a structured and efficient management of backend functionalities.

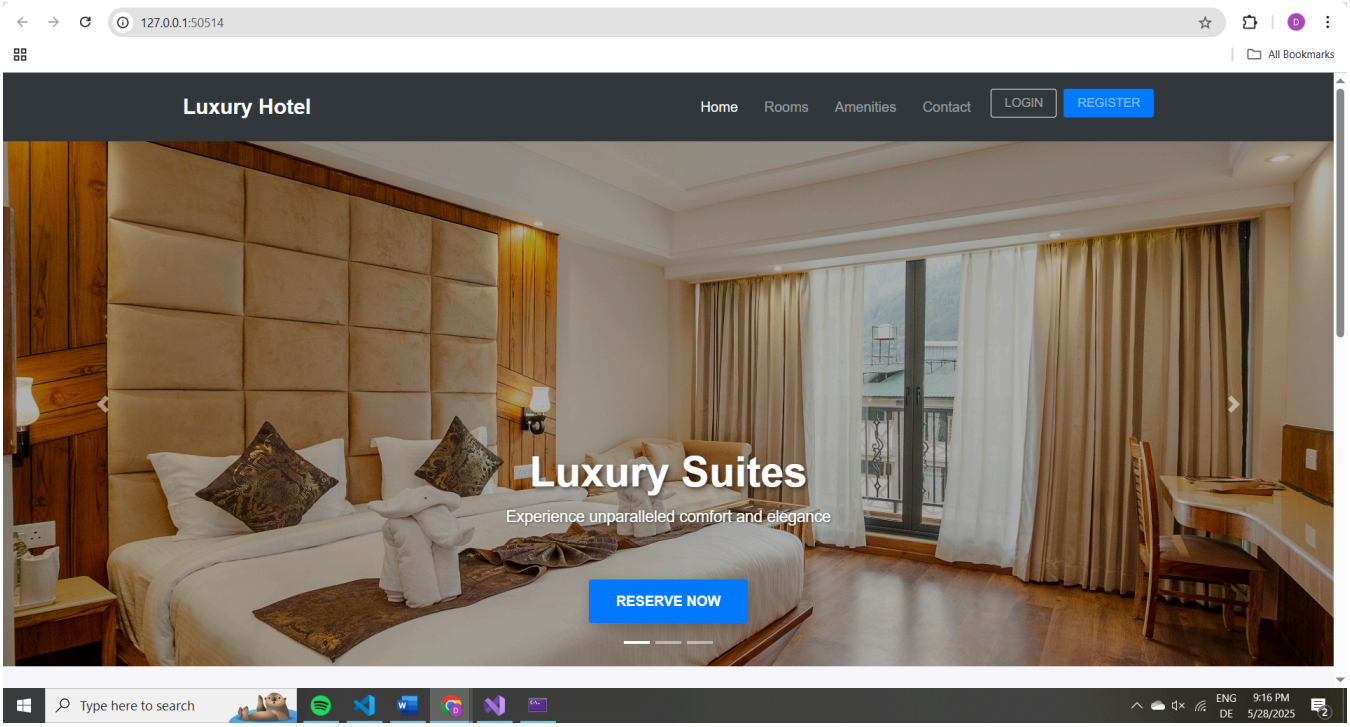
For the **Database**, we have used **Microsoft SQL Server**, and the database management is performed using **Microsoft SQL Server Management Studio (SSMS)**.

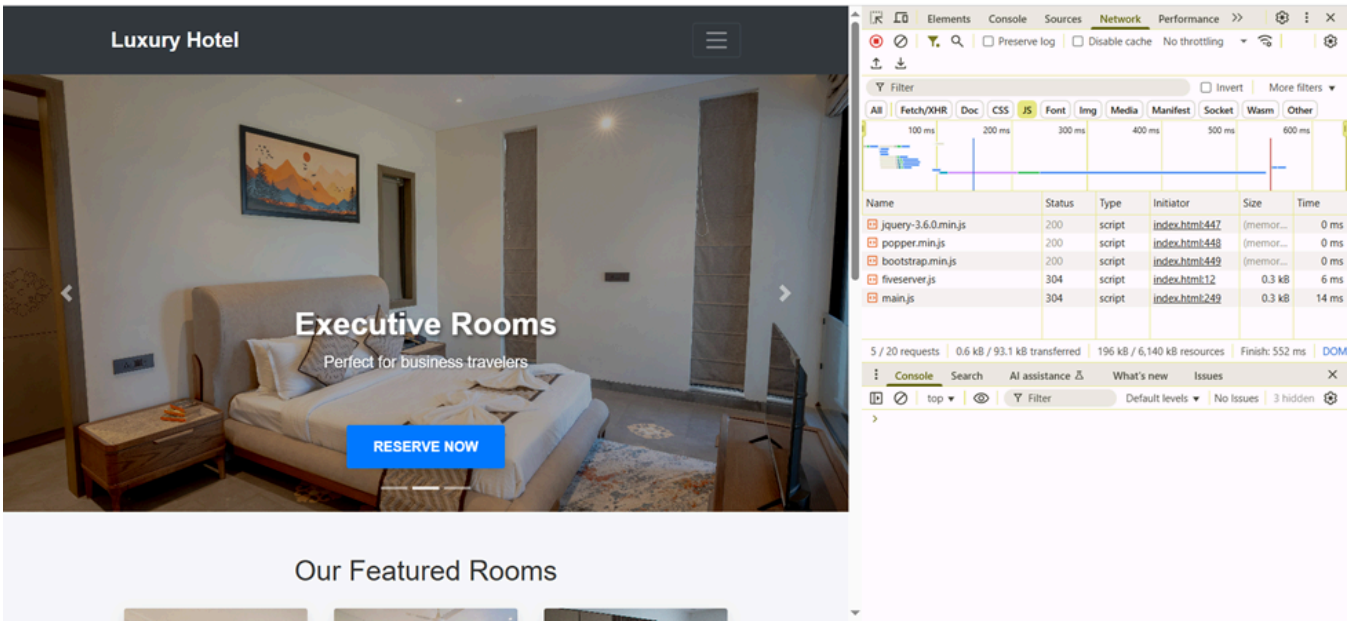
This project is also published in **GitHub**, where the step-by-step procedure of the creation of this project can be found (diagrams, designs, meeting reports, personal logs, etc.):

Github links:
https://github.com/LaertMema/HotelManagementSystemClient
https://github.com/LaertMema/HotelManagementSystem

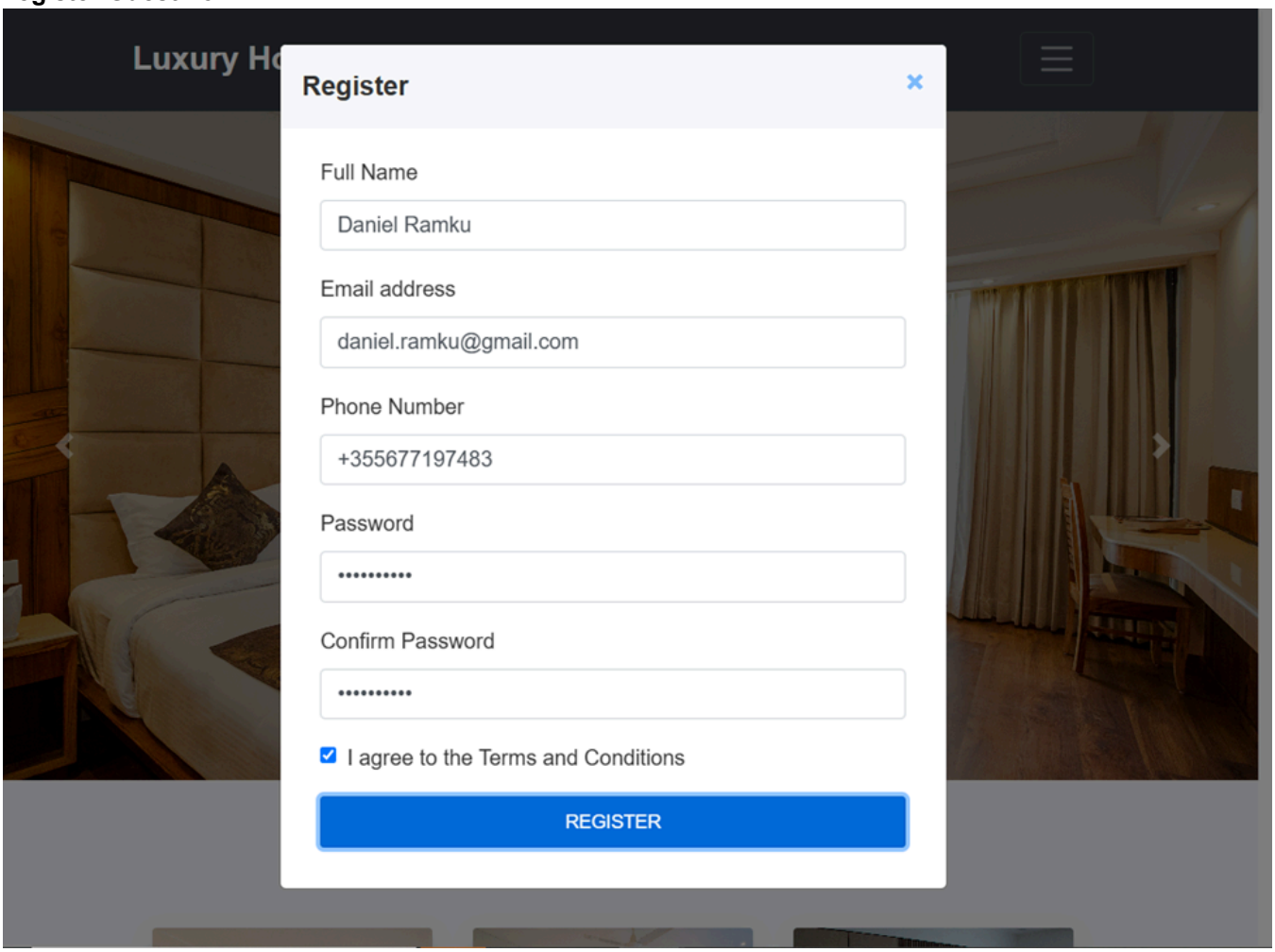
5.Appendix - Detailed Designs

Home Page

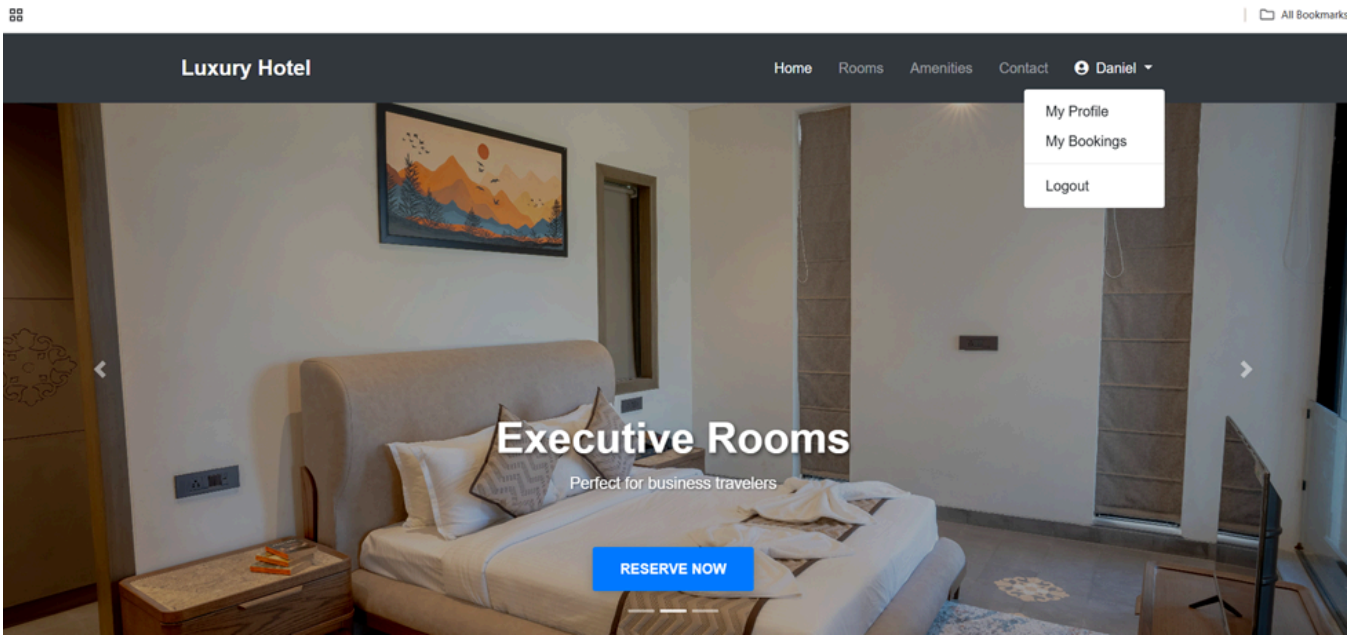




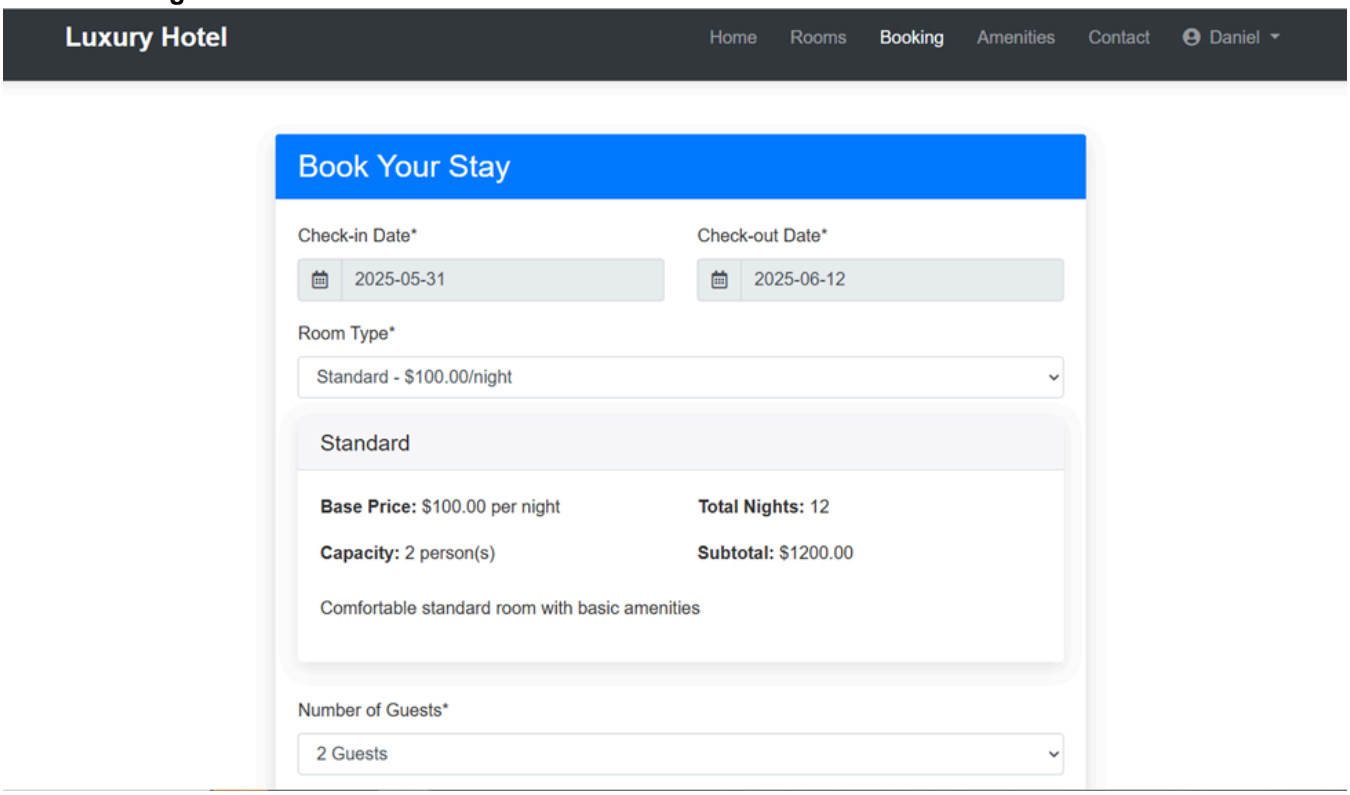
Register Guest Form



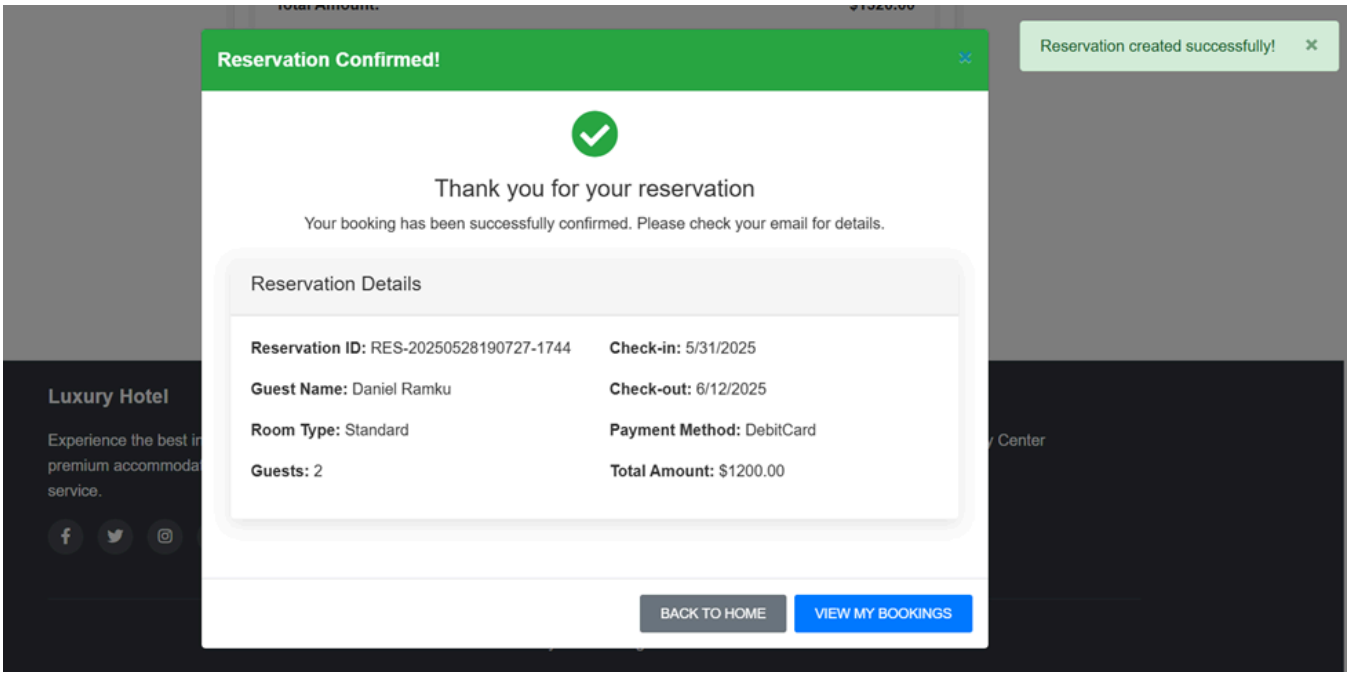
Log-In as Guest:



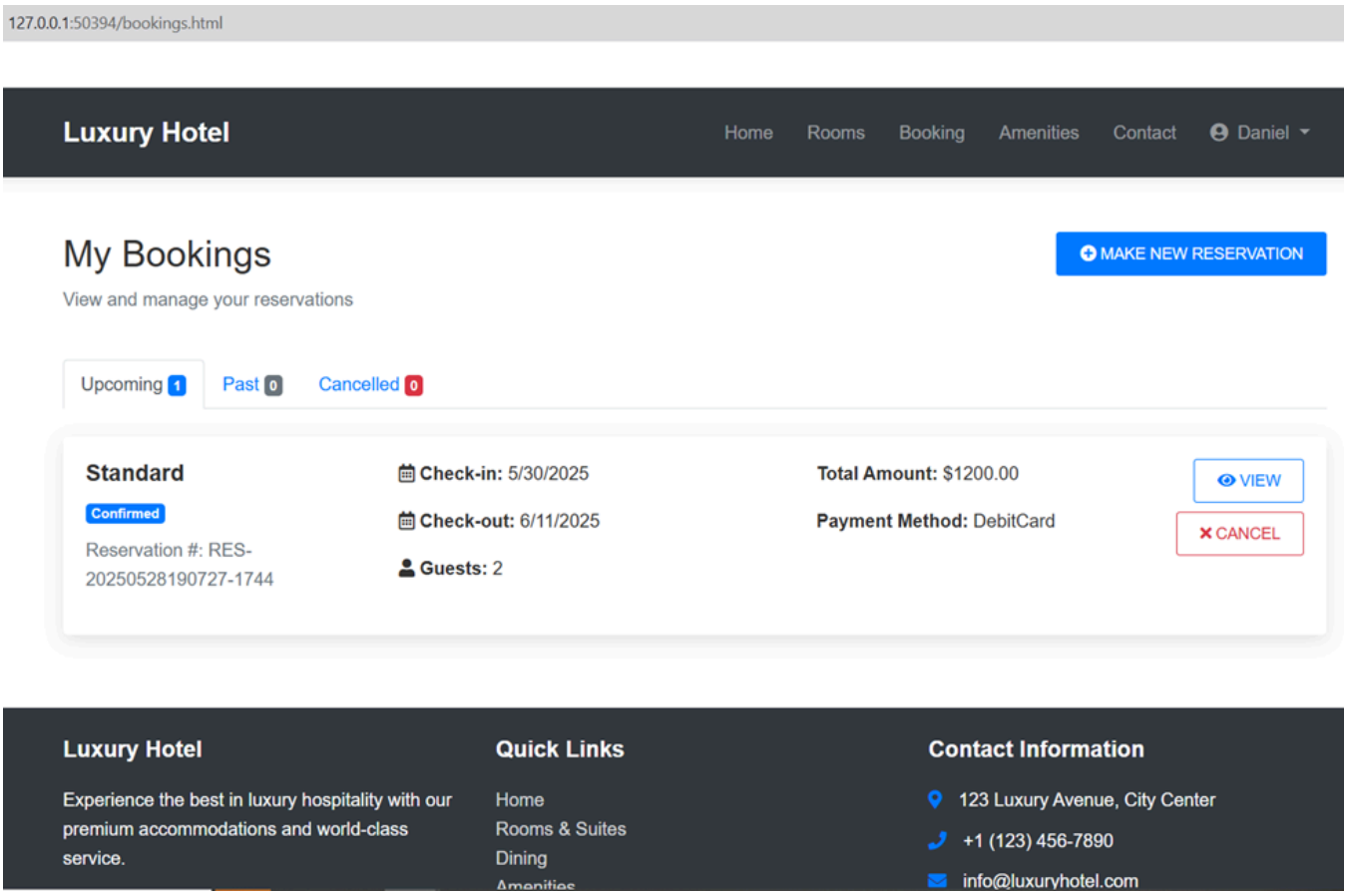
Guest booking a room



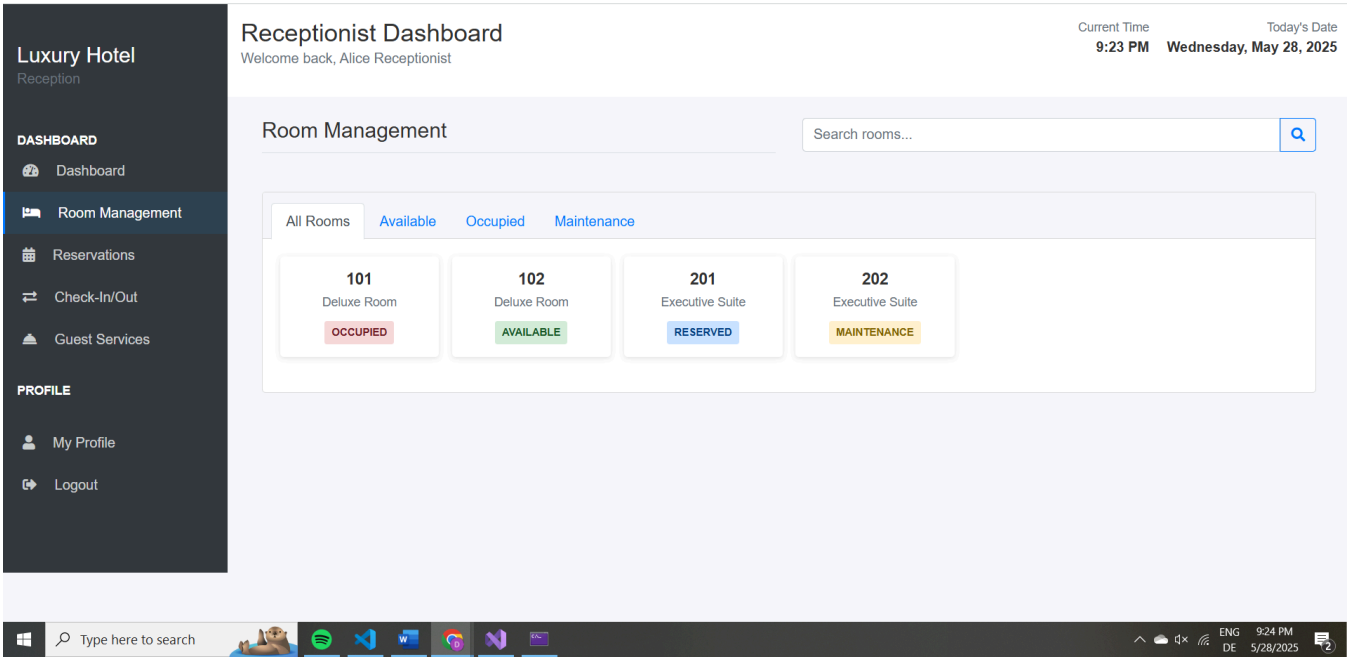
Reservation Confirmed



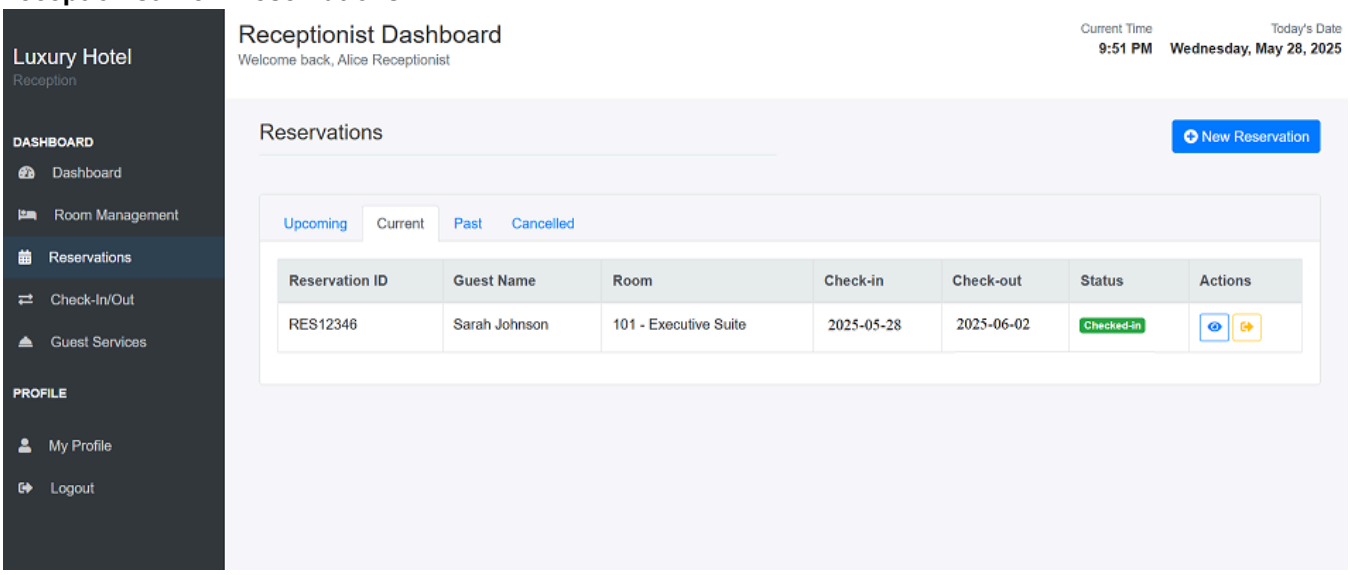
Guest View Bookings



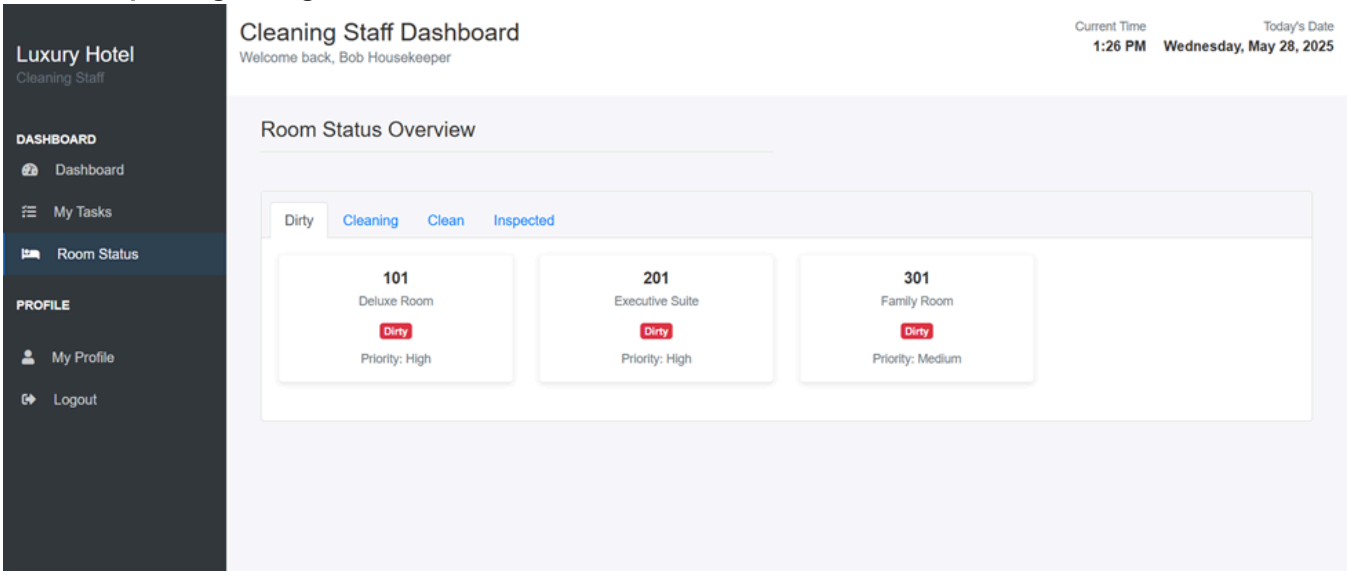
Receptionist Log-in Page



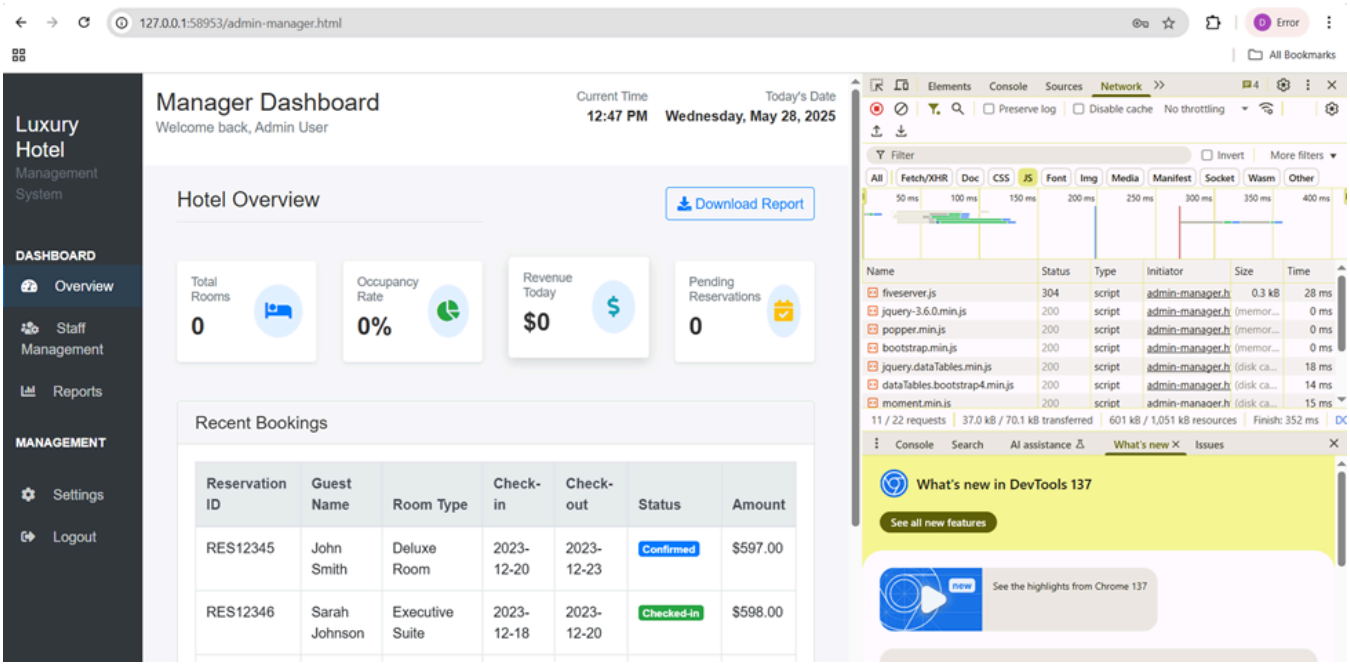
Receptionist View Reservations



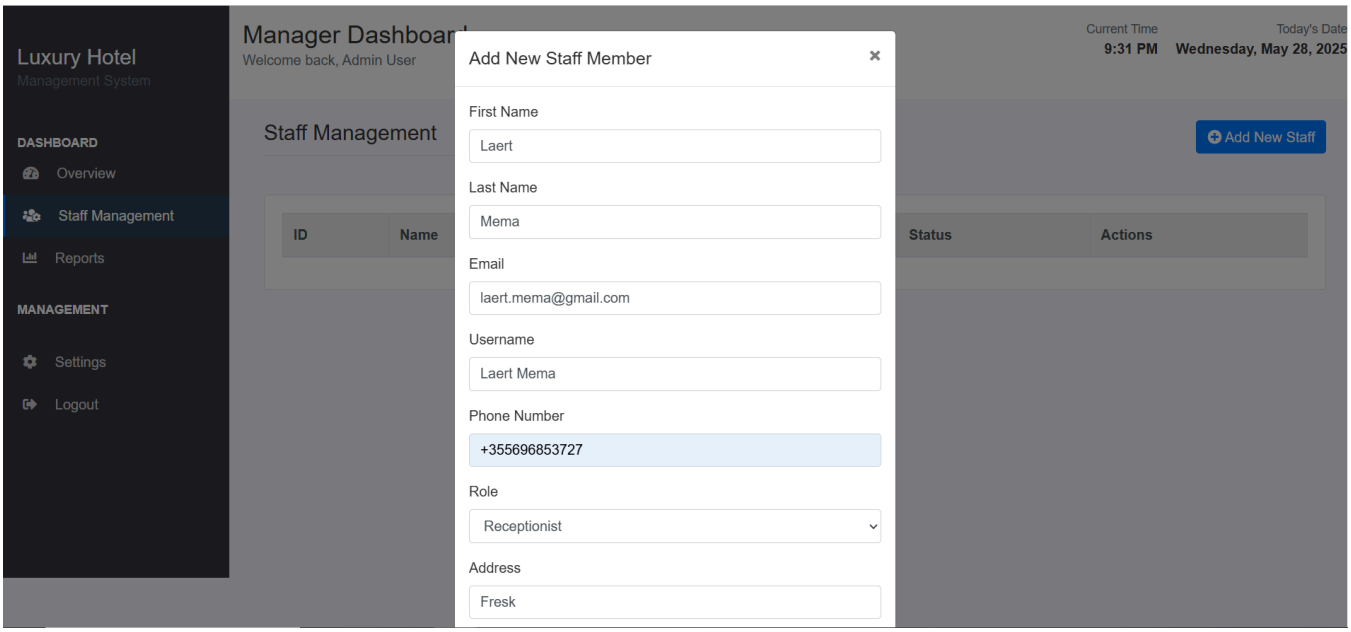
HouseKeeper Log-In Page



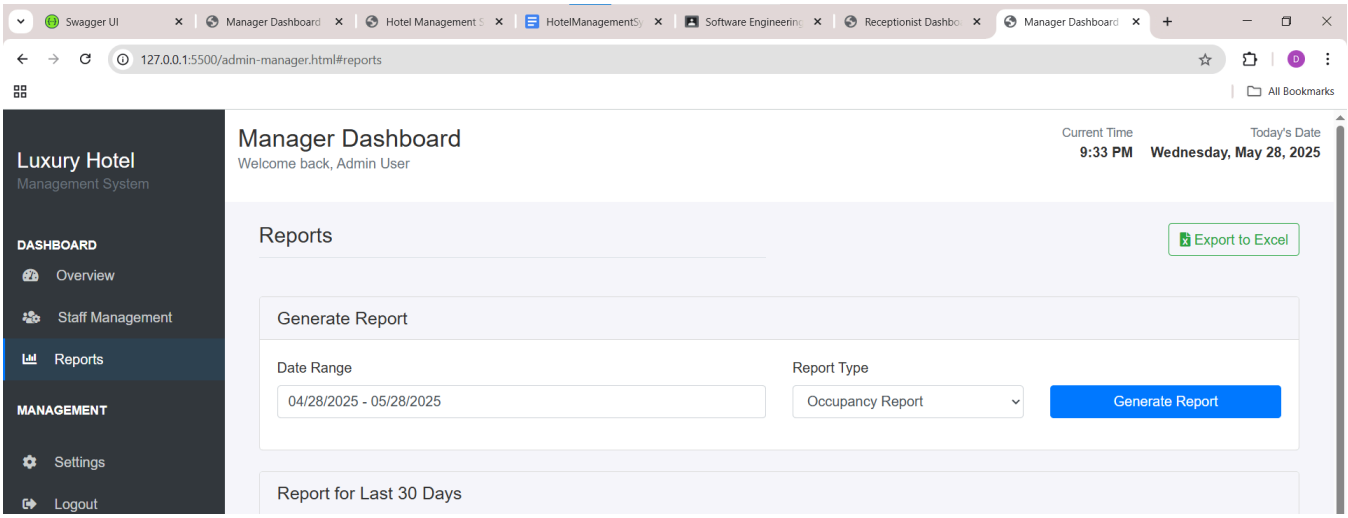
Admin Log-In Page



Admin-Add new Staff member



Admin Generate Report:



Manager Log-In page