Software Requirements and Design Document

For

HOSTEL HIVE

Prepared by

Muhammad Dawood (21i-0576)
Muhammad Osama Fahim (21i-0439)

Table of Contents

Contents

1. Introduction

1.1 Purpose

The purpose of the Hostel Management System (HMS) project is to streamline and enhance the overall management of hostel facilities. By providing a comprehensive solution, the HMS aims to optimize accommodation processes, automate billing and payment procedures, and improve the efficiency of meal management. This system is designed to cater to the diverse needs of hostel facilities, ensuring seamless operations and a positive experience for both administrators and hostel residents.

1.2 Product Scope

The product scope of the Hostel Management System (HMS) encompasses a range of functionalities and features designed to address the specific needs of hostel administrators, staff, and residents. The system aims to provide a comprehensive solution for efficient hostel management while ensuring user-friendly interfaces.

Key objectives and goals of the app include:

1. **User Management:** Administer user roles for administrators and hostelites with necessary authentication and authorization mechanism.
2. **Billing and Payment:** Automate payment process, generation of bills and securing online payment for meals and room services.
3. **Meal Management:** Providing a platform for Admins to fix rates for Breakfast, lunch and/or dinner and to choose whether they are providing all of three services or not so that hostelites can choose accordingly.
4. **Feedback and Improvement:** Collecting feedback from host elites and showing them to respective Admins to improve the services provided by Hostel Admins.

The software aligns with the corporate goals or business strategies by promoting engagement among Admins and hostelites so that Desired Hostel can be accommodated to the User which is a single click away.

1.3 Title

Hostel Hive: Unifying Accommodation Management for Seamless Hostel Living

1.4 Objectives

- **Efficient Accommodation Management :**
    - Choose your own Hostel according to desired location and facilities.
- **Enhance User Experience:**

- Provide a user-friendly interface for seamless registration, appointment scheduling, and interaction with Host elites and Admins.

- **Feedback and Improvement:**
  - If everything is not as it was promised you can always provide feedback which would be used.

1.5 Problem Statement

The Hostel Management System (HMS) project tackles key issues in hostel administration by offering an efficient and user-friendly solution. With a focus on meal management, administrators gain the ability to customize meal plans for hostels, deciding on breakfast, lunch, and dinner options. Simultaneously, hostel residents can access their bills, providing transparency in charges. Payment processes are seamlessly integrated, allowing residents to pay their bills and confirm their check-ins. The confirmation step involves administrator verification, ensuring a secure and controlled process. Additionally, the system facilitates room service requests initiated by residents, with administrators coordinating the response team. Residents can confirm the completion of room service tasks, enhancing communication and service quality. The feasibility of the HMS lies in its practicality, addressing specific hostel management challenges without unnecessary exaggeration.

2. Overall Description

2.1 Product Perspective

Hostel Hive emerges as a novel and independent solution tailored to address the unique requirements and challenges inherent in hostel management. It does not seek to replace current systems but instead offers an innovative approach to elevate and modernize the hostel management process. Hostel Hive is conceived as an independent platform, seamlessly integrating with existing hostel management structures and facilities. In a broader perspective, Hostel Hive stands as a pivotal element within the hostel management ecosystem. It interfaces with external systems such as notification services, location services, and databases to ensure streamlined communication, robust user engagement, and precise information retrieval. The platform is strategically designed to complement and augment the operational capabilities of hostel management organizations by providing a centralized, user-friendly interface for administrators and residents.

2.2 Product Functions

Following are the major functions of Hostel Hive:

User Registration and Login:

- Enable users, including hostel administrators and residents, to register for the platform by providing necessary information.

- Implement secure login processes for personalized access to the hostel management system.

Meal Management:

- Allow hostel administrators to define and manage meal plans, specifying which meals (breakfast, lunch, and dinner) will be provided to hostel residents.
- Provide residents with visibility into the meal plans, allowing them to view and plan their meals accordingly.

Bill Payment:

- Enable host elites to view their monthly bills, including room charges and meal expenses.
- Facilitate online payment options for residents to settle their bills securely.
- Room Service Requests:
- Allow residents to request room services through the platform, specifying their needs.
- Notify the hostel administration about service requests, and upon approval, update the resident about the scheduled service.

Check-In Confirmation:

- Initiate the check-in process for residents after successful payment confirmation by the admin.
- Ensure a seamless and efficient check-in experience for host elites.
- Feedback and Ratings:
- Collect feedback and ratings from hostel residents regarding various services and facilities.

2.3 List of Use Cases

1. Check-In Process
2. Feedback Collection Process
3. Room Maintenance
4. Billing Automation
5. Scheduled Maintenance
6. Inventory Tracking
7. Cash Handling and Reporting
8. Host elite Expense Tracking
9. Maintenance Request Management
10. Service Request Management

2.4 Extended Use Cases

"Check-In Process"

a. Use Case Name:

Check-In Process

b. Scope of the System under Design:

The system is designed to handle the check-in process for guests arriving at the hostel.

c. Level:

This is a primary use case at the highest level, involving the core functionality of the system.

d. Primary Actor:

Primary Actor: Guest

e. Stakeholders and Interests:

- Guest: Interested in a smooth and efficient check-in process.
- Hostel Management: Interested in ensuring a hassle-free check-in for guests to enhance their experience and collect necessary guest information.

f. Preconditions:

- The guest has made a reservation either online or through another channel.
- The guest has arrived at the hostel.

g. Post conditions:

- The guest is successfully checked into the hostel.
- Relevant information, such as guest details and room allocation, is updated in the system.

h. Main Success Scenario:

Actor's Actions	**System's Responsibilities**
The guest arrives at the hotel	Greet the guest and display a welcome message on the reception display.
The guest approaches the reception desk	Display available check-in options on the reception desk screen.
The guest provides identification (e.g., ID, reservation details) to the receptionist	Verify the guest's identification and reservation information for accuracy.

Actor's Actions	**System's Responsibilities**
The receptionist enters the guest's information into the check-in system	Record the guest's information accurately and securely in the system.
The system validates the guest's information, checks for available rooms, and verifies any special requests	Validate guest details, check room availability, and confirm any special requests or preferences.
The system assigns a room to the guest based on availability and preferences (e.g., room type, floor)	Automatically assign an available room that matches the guest's preferences, and update the room status as occupied.

I. Extensions:

- If the guest's reservation is not found, the system prompts the receptionist to check for errors or contact the guest's support.
- If the guest's payment is declined, the system prompts the receptionist to handle the payment issue.
- If the guest requests an upgrade or a change in room assignment, the system allows the receptionist to make the necessary adjustments.
- If the guest is arriving outside of regular check-in hours, the system may provide instructions for late check-in procedures.
- This use case outlines the process of checking in a guest at the hostel, with considerations for different scenarios and potential extensions to handle various situations efficiently.

"Feedback-Collection Process"

a. Use Case Name:

Feedback Collection

b. Scope of the System under Design:

The system is designed to collect feedback from hostel guests about their stay and experience.

c. Level:

This is a primary use case at the highest level, involving the core functionality of the system.

d. Primary Actor:

Primary Actor: Guest

e. Stakeholders and Interests:

- Guest: Interested in providing feedback about their experience.
- Hostel Management: Interested in gathering feedback to improve services and enhance the hostel's reputation.

f. Preconditions:

- The guest has completed their stay at the hostel.
- The guest has access to a device or system for providing feedback (e.g., mobile app, web portal).

g. Post conditions:

- The guest's feedback is successfully recorded in the system.
- Hostel management has access to feedback data for analysis and improvement.

h. Main Success Scenario:

Actor's Actions	**System's Responsibilities**
A resident wishes to provide feedback on their stay	Provide an accessible feedback submission interface.
The resident logs into their account (if required)	Authenticate the resident's credentials if login is required for feedback submission.
The resident navigates to the feedback section	Display a clearly labeled and easy-to-find feedback section.

Actor's Actions	**System's Responsibilities**
The resident selects the type of feedback (e.g., suggestion, complaint, compliment)	Present options for feedback type to categorize submissions.
The resident writes their feedback in a text box	Offer a text input field for the resident to compose their feedback.
The resident can optionally attach files (e.g., images, documents)	Allow residents to upload and attach files if needed.

I. Extensions:

- If the guest declines to provide feedback, the system acknowledges their decision and does not prompt further.
- If the guest encounters technical issues while submitting feedback, the system provides support or alternative methods for feedback submission.
- If the guest provides particularly positive feedback, the system may prompt them to leave a public review or share their experience on social media.
- If the guest provides negative feedback, the system may trigger an alert for hostel management to address the issue and follow up with the guest.

"Room Maintenance"

a. Use Case Name:

Room Maintenance

b. Scope of the System under Design:

The system is designed to manage and schedule maintenance tasks for hostel rooms to ensure they are in optimal condition for guests.

c. Level:

This is a primary use case at the highest level, involving the core functionality of the system.

d. Primary Actor:

Primary Actor: Maintenance Staff

e. Stakeholders and Interests:

- Maintenance Staff: Interested in receiving maintenance tasks, completing them efficiently, and updating the system with task status.
- Hostel Management: Interested in maintaining the hostel's physical assets to provide a positive guest experience.

f. Preconditions:

- Maintenance staff are logged into the HMS.
- Maintenance tasks are scheduled based on predefined maintenance intervals or guest reports.

g. Post conditions:

- Maintenance tasks are successfully completed and updated in the system.
- The hostel rooms are well-maintained and ready for guest occupancy.

h. Main Success Scenario:

Actor's Actions	**System's Responsibilities**
Hostel staff receives a maintenance request	Record and timestamp maintenance requests in the system.
Staff assesses the nature and urgency of the request	Prioritize maintenance tasks based on urgency and severity.
Staff assigns a maintenance team or technician	Assign the appropriate personnel to address the maintenance issue.

Actor's Actions	**System's Responsibilities**
The assigned team updates the maintenance status	Provide a platform for the team to update the status (e.g., in progress, completed).
The system notifies relevant parties of progress or completion	Automatically notify staff, residents, or management of maintenance progress or completion.
Maintain a maintenance log for auditing and historical data	Record details of completed maintenance tasks for future reference and auditing purposes.

I. Extensions:

- If the maintenance staff encounters an issue beyond their capabilities, the system allows them to escalate the task to a supervisor or specialist.
- If a guest reports a maintenance issue during their stay, the system may prioritize and assign the task to maintenance staff in real-time.
- If a maintenance task requires specific materials or tools, the system can facilitate inventory management to ensure necessary resources are available.

"Billing Automation"

a. Use Case Name:

Billing Automation

b. Scope of the System under Design:

The system is designed to automate the billing and invoicing process for guest stays and services within the hostel.

c. Level:

This is a primary use case at the highest level, involving the core functionality of the system.

d. Primary Actor:

Primary Actor: Receptionist or Automated Billing System

e. Stakeholders and Interests:

- Receptionist: Interested in efficiently generating accurate bills for guest stays and services.
- Guests: Interested in receiving clear and accurate bills for their stay and any additional services.
- Hostel Management: Interested in streamlining the billing process and minimizing billing errors.

f. Preconditions:

- The guest has completed their stay or used hostel services (e.g., laundry, meals).
- All relevant charges and rates are defined and updated in the system.

g. Post conditions:

- A detailed and accurate bill is generated for the guest's stay, including any additional services.
- The guest receives a clear invoice, and charges are recorded in the system.

h. Main Success Scenario:

Actor's Actions	**System's Responsibilities**
The system identifies guest check-out or service usage.	Automatically track guest activities and services.
The system calculates charges based on predefined rates.	Accurately compute charges for room and services.
Generate a detailed invoice with charge breakdown.	Create a clear invoice for guest stay and services.
Present the invoice to the guest during check-out.	Display the invoice for guest review and payment.
Resolve any billing discrepancies or questions.	Allow guests to inquire about or dispute charges.

I. Extensions:

- If there are disputes or questions regarding charges, the system allows the receptionist to investigate and make necessary adjustments.
- If a guest has a special rate, discount, or voucher, the system can apply these discounts during billing.
- If the guest prefers to receive the invoice electronically, the system can send it via email.

"Scheduled Maintenance"

a. Use Case Name:

Scheduled Maintenance

b. Scope of the System under Design:

The system is designed to manage and schedule routine maintenance and upgrades for various areas and facilities within the hostel.

c. Level:

This is a primary use case at the highest level, involving the core functionality of the system.

d. Primary Actor:

Primary Actor: Maintenance Manager

e. Stakeholders and Interests:

- Maintenance Manager: Interested in planning and executing scheduled maintenance tasks efficiently.
- Hostel Management: Interested in ensuring the long-term maintenance and improvement of hostel facilities.

f. Preconditions:

- A maintenance schedule is defined, specifying when and which areas or facilities require maintenance.
- Maintenance staff and resources are available and allocated.

g. Post conditions:

- Scheduled maintenance tasks are successfully completed.
- Hostel facilities are well-maintained and continue to meet quality standards.

h. Main Success Scenario:

Actor's Actions	**System's Responsibilities**
Hostel staff initiates a maintenance request	Record and categorize maintenance requests.
Specify the type and urgency of maintenance needed	Prioritize and schedule maintenance tasks.
Requestor assigns the maintenance task to a staff member or team	Allocate tasks to available maintenance personnel.
Maintenance staff access their schedules and task assignments	Provide staff with access to their assigned tasks and schedules.
Staff update task status (e.g., in progress, completed)	Allow staff to update and track task progress.
Generate reports on maintenance performance	Create reports for management to assess maintenance efficiency and quality.

I. Extensions:

- If a maintenance task encounters unexpected issues or requires additional resources, the system can facilitate adjustments to the schedule or resource allocation.
- If certain areas of the hostel require more frequent or specialized maintenance, the system can allow for customization of maintenance schedules.

- If maintenance tasks are not completed within the scheduled time frame, the system can trigger notifications and reminders to ensure timely completion.

"Inventory Tracking"

a. Use Case Name:

Inventory Tracking

b. Scope of the System under Design:

The system is designed to track and manage the inventory of supplies, consumables, and equipment used within the hostel.

c. Level:

This is a lower-level use case that supports the higher-level "Efficient Hostel Operations" use case by ensuring that essential supplies are always available.

d. Primary Actor:

Primary Actor: Inventory Manager

e. Stakeholders and Interests:

- Inventory Manager: Interested in maintaining optimal inventory levels, reducing wastage, and ensuring supplies are always available when needed.
- Hostel Management: Interested in cost-effective inventory management to support hostel operations.

f. Preconditions:

- The inventory manager is logged into the HMS.
- Inventory levels have been defined and updated in the system.

g. Post conditions:

- Inventory levels are accurately tracked, and reorder alerts are generated when supplies reach predefined thresholds.

h. Main Success Scenario:

Actor's Actions	**System's Responsibilities**
Hostel staff records the receipt of new inventory items	Log new inventory items into the system with details like quantity, description, and date received.

Actor's Actions	**System's Responsibilities**
Staff updates inventory levels as items are used or consumed	Track inventory levels in real-time as items are used or replenished.
The system automatically generates low inventory alerts	Monitor inventory levels and notify staff when items are running low.
Staff can request restocking of low inventory items	Allow staff to initiate restocking requests when inventory levels fall below a certain threshold.
The system tracks the status of restocking requests	Keep a record of restocking requests, including approval status and delivery timelines.
Generate reports on inventory usage and status	Create reports for management to analyze inventory trends, usage, and expenses.

I. Extensions:

- If certain items are prone to seasonal demand variations, the system allows for adjusting reorder points and quantities.
- If a specific item is consistently overstocked, the system can provide recommendations for adjusting procurement quantities.
- If there are issues with the quality or condition of received supplies, the system allows for recording and tracking such incidents.

"Host elite Expense Tracking"

a. Use Case Name:

Host elite Expense Tracking

b. Scope of the System under Design:

- The system is designed to allow individual hostel residents (hostilities) to track and
- Manage their personal expenses during their stay in the hostel.

c. Level:

- This is a lower-level use case that contributes to the higher-level "Improved Guest Experience" use case by
- Enabling hostilities to manage their finances efficiently.

d. Primary Actor:

Primary Actor: Host elite (Guest)

e. Stakeholders and Interests:

- Host elites: Interested in keeping a record of their expenses and managing their budgets during their hostel stay.
- Hostel Management: Interested in providing a helpful feature for hostelites to enhance their experience.

f. Preconditions:

- The host elite has an account in the HMS.
- The host elite is logged into the system.

g. Post conditions:

- The host elite successfully records and tracks their personal expenses while staying at the hostel.

h. Main Success Scenario:

Actor's Action	**System Responsibility**
Host logs into the system.	Authenticates the host's credentials and grants access to the dashboard.
Host creates new expense categories.	Provides an interface for category creation and stores the information securely.

Actor's Action	**System Responsibility**
Host invites team members.	Generates and sends invitations to team members, creating user accounts when they accept.
Host and team members log expenses.	Offers an input form and stores submitted expense data securely in the database.
Host generates expense reports.	Provides tools for report generation and customization based on the stored data.
Host and team members export reports.	Allows for the export of reports in different formats (e.g., PDF, CSV) for accounting purposes.

I. Extensions:

- If the host elite wishes to split expenses with roommates or friends, the system can support expense sharing and bill splitting features.
- If the host elite wants to export their expense data for personal financial management, the system allows for data export in various formats (e.g., CSV, PDF).
- If there are changes in expense categories or reporting preferences, the system can be updated to accommodate these changes.

"Maintenance Request Management"

a. Use Case Name:

Maintenance Request Management

b. Scope of the System under Design:

The system is designed to manage and streamline the process of receiving, prioritizing, and addressing maintenance requests from hostel residents.

c. Level:

This is a lower-level use case that contributes to the higher-level "Efficient Hostel Operations" use case by ensuring that maintenance requests are handled promptly.

d. Primary Actor:

Primary Actor: Hostel Resident

e. Stakeholders and Interests:

- Hostel Resident: Interested in reporting maintenance issues in their rooms or common areas and having them resolved efficiently.
- Maintenance Staff: Interested in receiving and prioritizing maintenance requests for timely resolution.
- Hostel Management: Interested in maintaining the overall condition and safety of the hostel.

f. Preconditions:

- The hostel resident is logged into their account in the HMS.
- Maintenance categories and request handling procedures are defined in the system.

g. Post conditions:

- Maintenance requests are successfully submitted, tracked, and resolved.

h. Main Success Scenario:

Actor's Action	**System Responsibility**
User submits a maintenance request.	Receives and records the maintenance request details, including the issue description and location.
Maintenance team receives the request.	Notifies the maintenance team about the new request and assigns it to an available team member.

Actor's Action	**System Responsibility**
Maintenance team member reviews the request.	Accesses the request details, assesses the issue, and schedules a visit if required.
Maintenance team member updates the request status.	Updates the request status, such as "In Progress," "Completed," or "Requires Further Attention."
User receives notification of the status change.	Sends automated notifications to the user regarding the status change and any additional information.
User provides feedback on the completed request.	Allows the user to provide feedback on the maintenance service, which is recorded for future improvement.

I. Extensions:

- If a maintenance request is urgent or poses a safety hazard, the system can escalate it for immediate attention.
- If the maintenance staff encounter challenges during the task (e.g., lack of required parts), the system can facilitate communication with hostel management or procurement.
- If the resident is dissatisfied with the resolution, they can provide feedback or request a follow-up.

"Meal Ordering and Delivery"

a. Use Case Name:

Meal Ordering and Delivery

b. Scope of the System under Design:

The system is designed to allow hostel residents to order meals from the hostel's dining facilities and have them delivered to their rooms or designated dining areas.

c. Level:

This is a lower-level use case that contributes to the higher-level "Improved Guest Experience" use case by enhancing the convenience of dining services for hostel residents.

d. Primary Actor:

Primary Actor: Hostel Resident

e. Stakeholders and Interests:

- Hostel Resident: Interested in ordering meals conveniently and having them delivered to their preferred location.
- Dining Staff: Interested in receiving meal orders, preparing meals, and ensuring timely delivery.
- Hostel Management: Interested in providing efficient dining services to enhance the guest experience and revenue.

f. Preconditions:

- The hostel resident is logged into their account in the HMS.
- Dining menu options, pricing, and delivery logistics are defined in the system.

g. Post conditions:

- The hostel resident successfully places a meal order, and the dining staff prepares and delivers the meal as requested.

h. Main Success Scenario:

Actor's Action	**System Responsibility**
User selects and customizes a meal order.	Provides a user-friendly menu interface with meal options and customization features. Records the order details.
User confirms the order and makes payment.	Validates the order information and securely processes the payment, providing a confirmation receipt.
Restaurant receives the order and prepares it.	Notifies the restaurant staff about the incoming order and displays order details for preparation.

Actor's Action	**System Responsibility**
Delivery driver is assigned to pick up the order.	Assigns a delivery driver, shares order details, and provides estimated delivery time to the driver.
User receives the meal at the specified location.	Updates the user on the delivery status and provides real-time tracking information until the meal is delivered.
User provides feedback on the delivered meal.	Allows the user to provide feedback on the meal and overall experience, which can be used for quality control.

I. Extensions:

- If there are delays or changes to the delivery time, the system can provide real-time updates to the resident.
- If the resident wishes to modify or cancel the order, the system allows for order management up to a certain cutoff time.
- If the dining staff encounters issues with meal availability or special requests, they can communicate with the resident through the system.

"Service Request Management"

a. Use Case Name:

Service Request Management

b. Scope of the System under Design:

The system is designed to manage and streamline various service requests made by hostel residents, including laundry, maintenance, housekeeping, and other services.

c. Level:

This is a lower-level use case that contributes to the higher-level "Improved Guest Experience" use case by providing, a unified and efficient platform for residents to request services.

d. Primary Actor:

Primary Actor: Hostel Resident

e. Stakeholders and Interests:

- Hostel Resident: Interested in requesting various services to enhance their stay and resolve issues promptly.
- Service Staff: Interested in receiving and managing service requests efficiently to meet resident needs.
- Hostel Management: Interested in providing responsive and high-quality services to enhance guest satisfaction.

f. Preconditions:

- The hostel resident is logged into their account in the HMS.
- Service options, availability, and scheduling parameters are defined in the system.

g. Post conditions:

- The hostel resident successfully places a service request, and the appropriate staff addresses the request in a timely manner.

h. Main Success Scenario:

Actor's Action	**System Responsibility**
User submits a service request.	Receives and records the service request details, including the issue description and priority level.
Service team receives the request.	Notifies the service team about the new request and assigns it to an available team member.
Service team member reviews the request.	Accesses the request details, assesses the issue, and plans a response, including necessary resources.

Actor's Action	**System Responsibility**
Service team member updates the request status.	Updates the request status, such as "In Progress," "Completed," or "Requires Further Attention."
User receives notification of the status change.	Sends automated notifications to the user regarding the status change and any additional information.
User provides feedback on the completed request.	Allows the user to provide feedback on the service received, which is recorded for continuous improvement.

I. Extensions:

- If there are delays or changes to the service delivery time, the system can provide real-time updates to the resident.
- If the resident wishes to modify or cancel the service request, the system allows for request management up to a certain cutoff time.
- If there are issues or questions related to the service request, the resident and service staff can communicate through the system.

"Cash Handling and Reporting"

a. Use Case Name:

Cash Handling and Reporting

b. Scope of the System under Design:

The system is designed to manage and track cash transactions within the hostel, including cash handling at the front desk, recording cash income, and generating financial reports.

c. Level:

This is a lower-level use case that contributes to the higher-level "Cost Reduction" and "Efficient Hostel Operations" use cases by ensuring accurate financial tracking and reducing the risk of cash mishandling.

d. Primary Actor:

Primary Actor: Front Desk Receptionist or Cashier

e. Stakeholders and Interests:

- Front Desk Receptionist or Cashier: Interested in accurately recording cash transactions, reconciling cash drawers, and generating financial reports.
- Hostel Management: Interested in maintaining accurate financial records, minimizing errors, and optimizing cash management processes.

f. Preconditions:

- The front desk receptionist or cashier is logged into their account in the HMS.
- Cash handling procedures and reconciliation guidelines are defined in the system.

g. Post conditions:

- Cash transactions are accurately recorded, cash drawers are reconciled, and financial reports are generated.

h. Main Success Scenario:

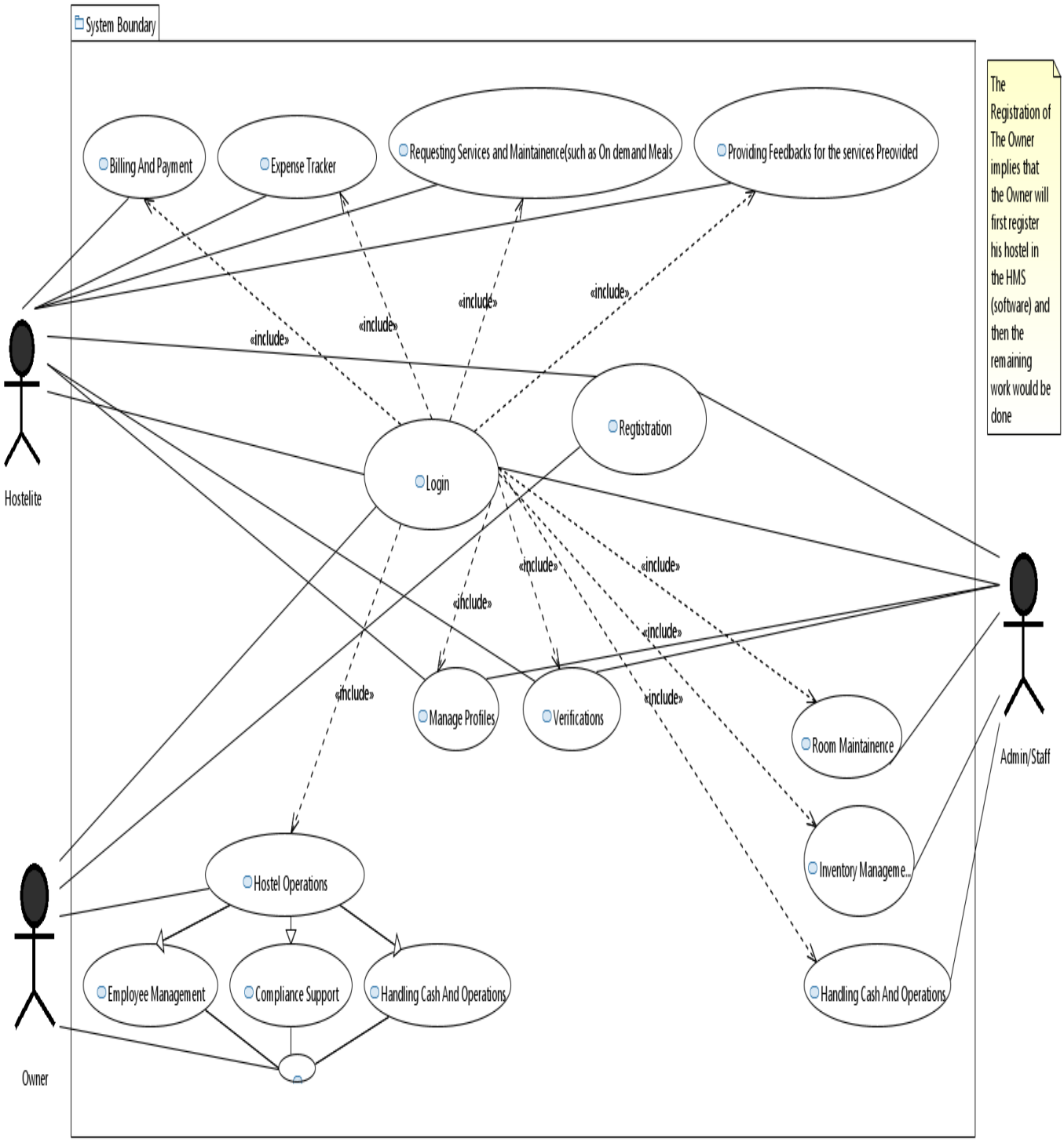
Actor's Action	**System Responsibility**
Cashier receives cash payments.	Records the cash transactions, including the amount received and the purpose (e.g., sales, deposits).
Cashier updates the cash register balance.	Maintains an accurate running balance of cash in the register, factoring in both receipts and disbursements.
Manager generates cash reports.	Provides tools for managers to generate cash reports, summarizing daily or periodic cash transactions.
System reconciles cash transactions.	Automatically reconciles the recorded transactions with the register balance, identifying any discrepancies.

Actor's Action	**System Responsibility**
Manager reviews and approves cash reports.	Allows managers to review and validate the generated cash reports, ensuring accuracy and compliance.
Approved reports are stored for record-keeping.	Archives approved cash reports securely for auditing purposes, maintaining a historical record of transactions.

I. Extensions:

- If there are significant discrepancies in cash drawer reconciliation, the system can trigger notifications or alerts for further investigation.
- If the receptionist or cashier needs to record cash expenses (e.g., petty cash expenditures for small purchases), the system allows for expense recording.

2.5 Use Case diagram



1. Other Nonfunctional Requirements

a. Performance Requirements

Responsiveness:

- The app should provide a responsive user interface, ensuring users experience minimal delays during interactions.

Scalability:

- The app should be scalable to accommodate potential growth in the user base, ensuring consistent performance during periods of increased usage.

Scheduled Maintenance Efficiency:

- The systems scheduled maintenance tasks should be performed efficiently, minimizing downtime and disruptions to hostel operations.

Efficient Billing Automation:

- Billing automation should be optimized for speed and accuracy, ensuring swift processing of host elite bills.

b. Safety Requirements

- **Secure Authentication:**
  - The app must implement secure authentication mechanisms to prevent unauthorized access to user accounts, ensuring that only legitimate users can interact with sensitive features.
- **Room Service Requests Authentication:**
  - Hostelites will be able to request room services and a team will be sent to that room by the Admin and after service has been completed successfully, hostelite will click on confirm button.
- **Bill Automation:**
  - Bills will be paid by hostelites and after that the transaction will be confirmed by the respective hostel admin.

c. Security Requirements

1. **User Authentication:**
   - Users must undergo secure authentication processes to access the app, ensuring that only authorized individuals can interact with sensitive functionalities.
2. **Secure Storage:**

- User data, including personal information and health details, must be securely stored, following industry best practices for data security.

3. **Privacy Settings:**
   - Users should have control over their privacy settings, allowing them to manage the visibility of their profiles and personal information to other users.
4. **Regular Security Audits:**
   - Conduct regular security audits to identify and address potential vulnerabilities, ensuring the ongoing security of the app and its infrastructure.
5. **Incident Response Plan:**
   - Develop and maintain an incident response plan to address security incidents promptly, minimizing the impact on users and the integrity of the app.

d. Software Quality Attributes

1. **Interoperability:**
   - **Requirement:** The app should be compatible with most of commonly used mobile devices.
   - **Reason:** Interoperability ensures a broader reach and accessibility for users across different devices.
2. **Scalability:**
   - **Requirement:** The app should handle an increase in concurrent users without significant performance degradation.
   - **Reason:** Scalability is crucial to accommodate potential growth in the user base without compromising performance.
3. **Flexibility:**
   - **Requirement:** The hostel services will improved according to the feedbacks given to the admins with immediate effect.
   - **Reason:** Flexibility enables the continuous improvement of the app without causing inconvenience to users.

e. Business Rules

- **Admins** will be able to effectively and accurately list hostels, provide meal descriptions regarding their hostels i.e. Which meal is provided by their hostel or not (breakfast, lunch, or dinner), View Feedbacks to take necessary actions, confirm Bill payments indicating that payment has been received and hostelite will be checked in and send Teams for room services.

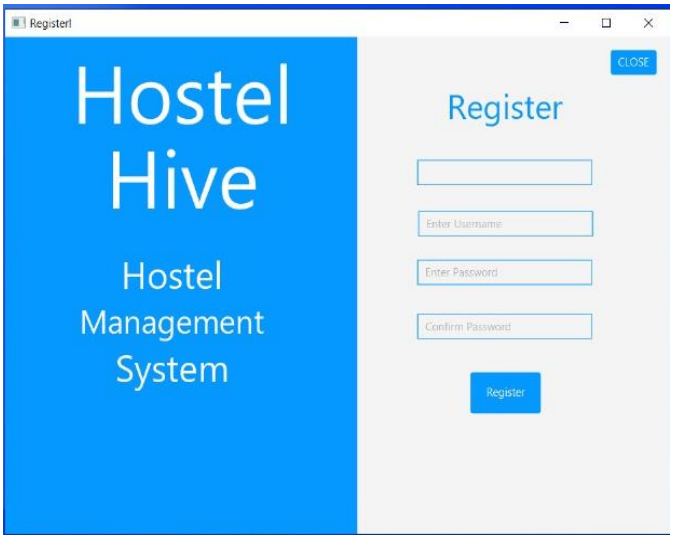
- **Hostelites** will be able to reserve desired hostel, manage expenses, give feedbacks regarding their hostels, confirm room services indicating that team has arrived for inspection and further maintenance and hostelite will be able to pay bills with a confirmation from respective hostel admin indicating secure transactions.

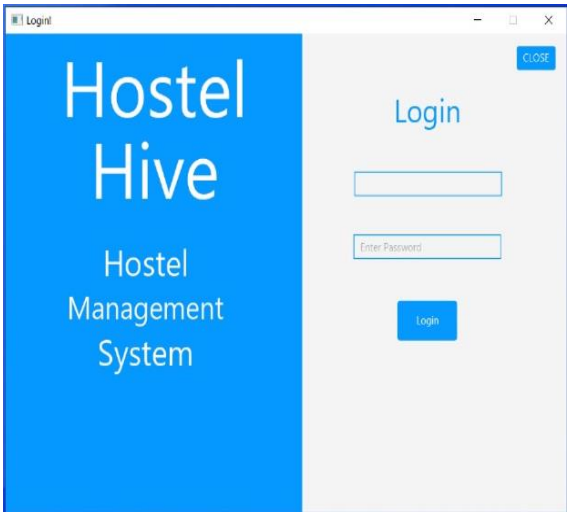
f. Operating Environment

- **Hardware Platform:**
  - The app is intended to operate on PCs and laptops.
- **Operating System:**
  - The app is compatible with windows operating systems.
- **Software Components:**
  - The app interacts with external servers for data storage, retrieval, and communication.
- **Database:**
  - The app relies on a secure and scalable database for storing user profiles, donation history, and other relevant information.
  - The database is part of the overall system architecture and ensures data integrity and accessibility.
- **Security Features:**
  - The app incorporates security features to protect user data, including secure storage practices.
  - It may utilize secure authentication mechanisms for user access.

g. User Interfaces

1. **Registration/Login Interface:**
   - **Description:** Users access the registration/login interface upon launching the app for the first time.
   - **Elements:** Input fields for essential information (name, email, etc.), confirmation messages, and a secure login interface.

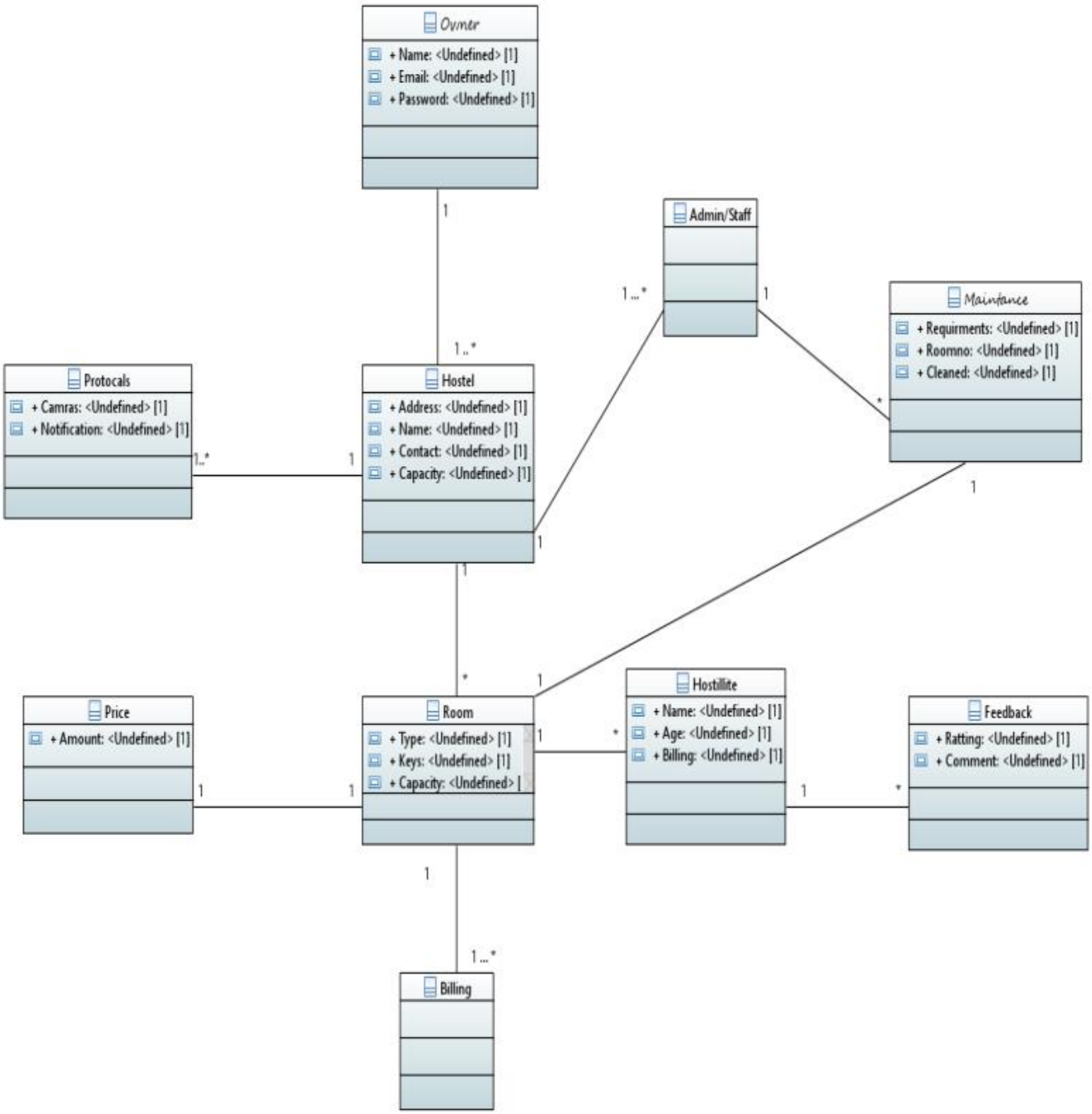




2. **Home Screen:**
   - **Description:** The main screen where users land after login, providing an overview of key app features.
   - **Elements:** Personalized features such as Reservation, Bill Automation, and Feedback.

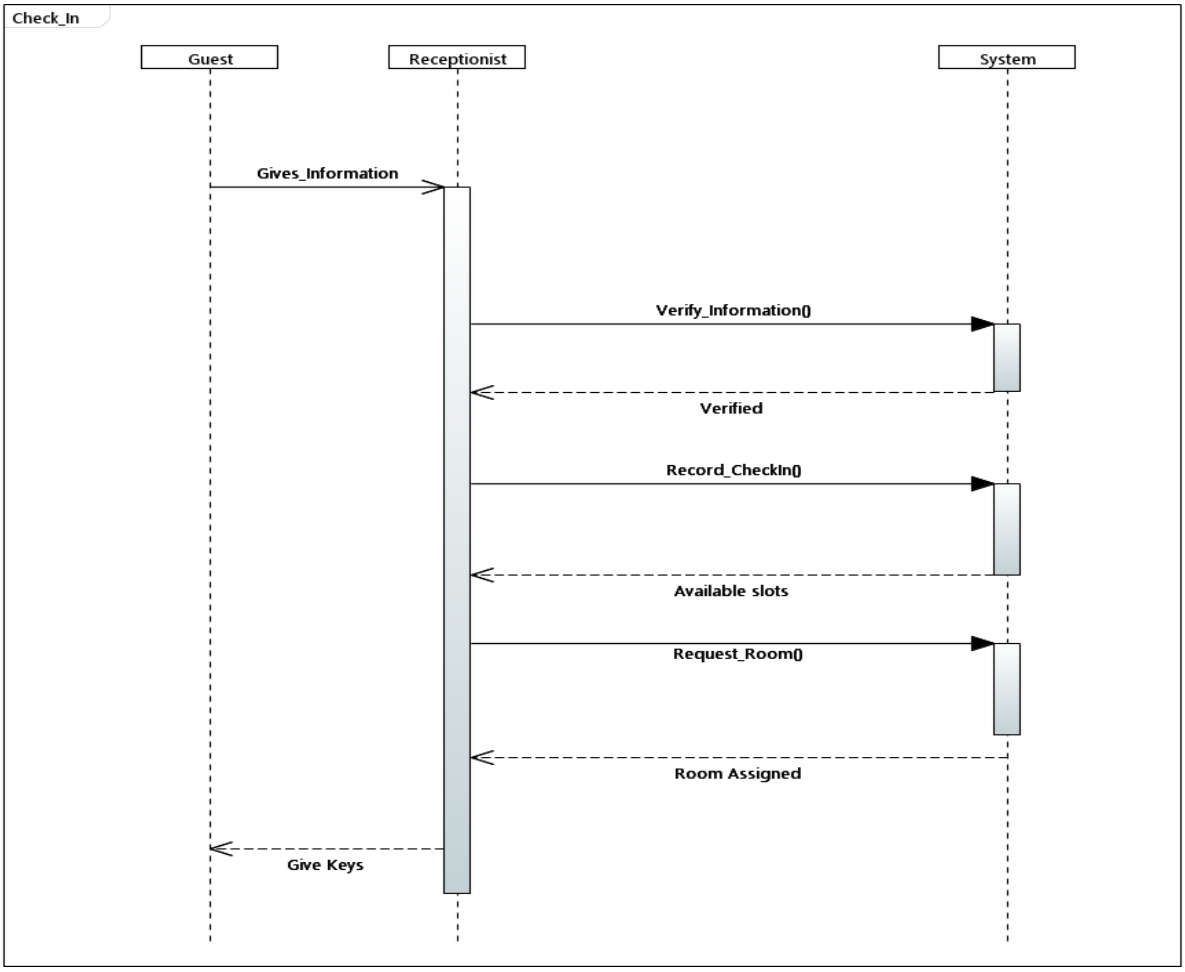
3. **Room Service Interface:**
    - **Description:** Users can schedule appointments for room services.
    - **Elements:** Users can request room service and a team will be sent for service and after that host elite will click on confirm button to confirm that the team had arrived and service is done successfully.

2. Domain Model

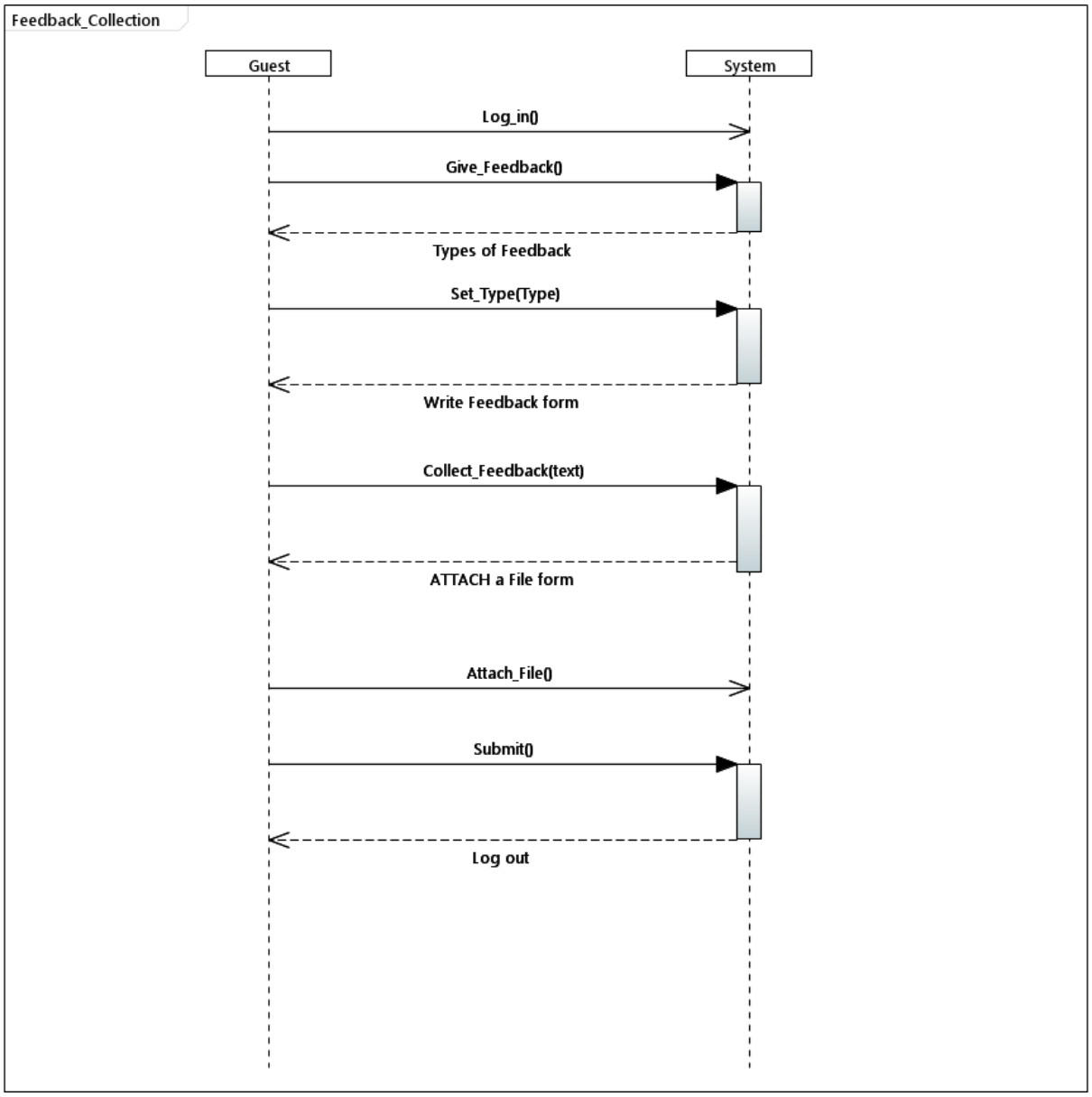


3. System Sequence Diagram

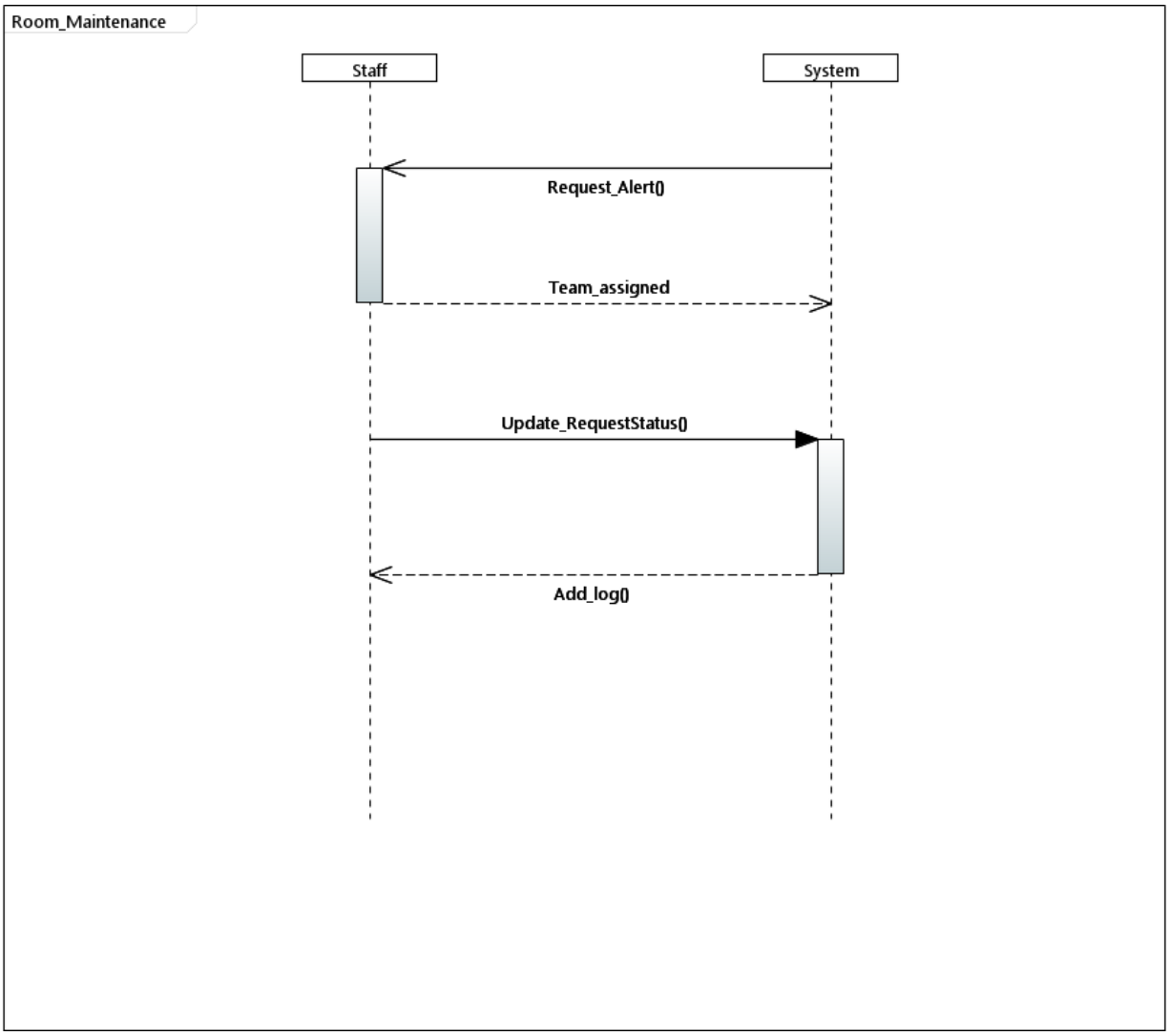
1. Check-In:



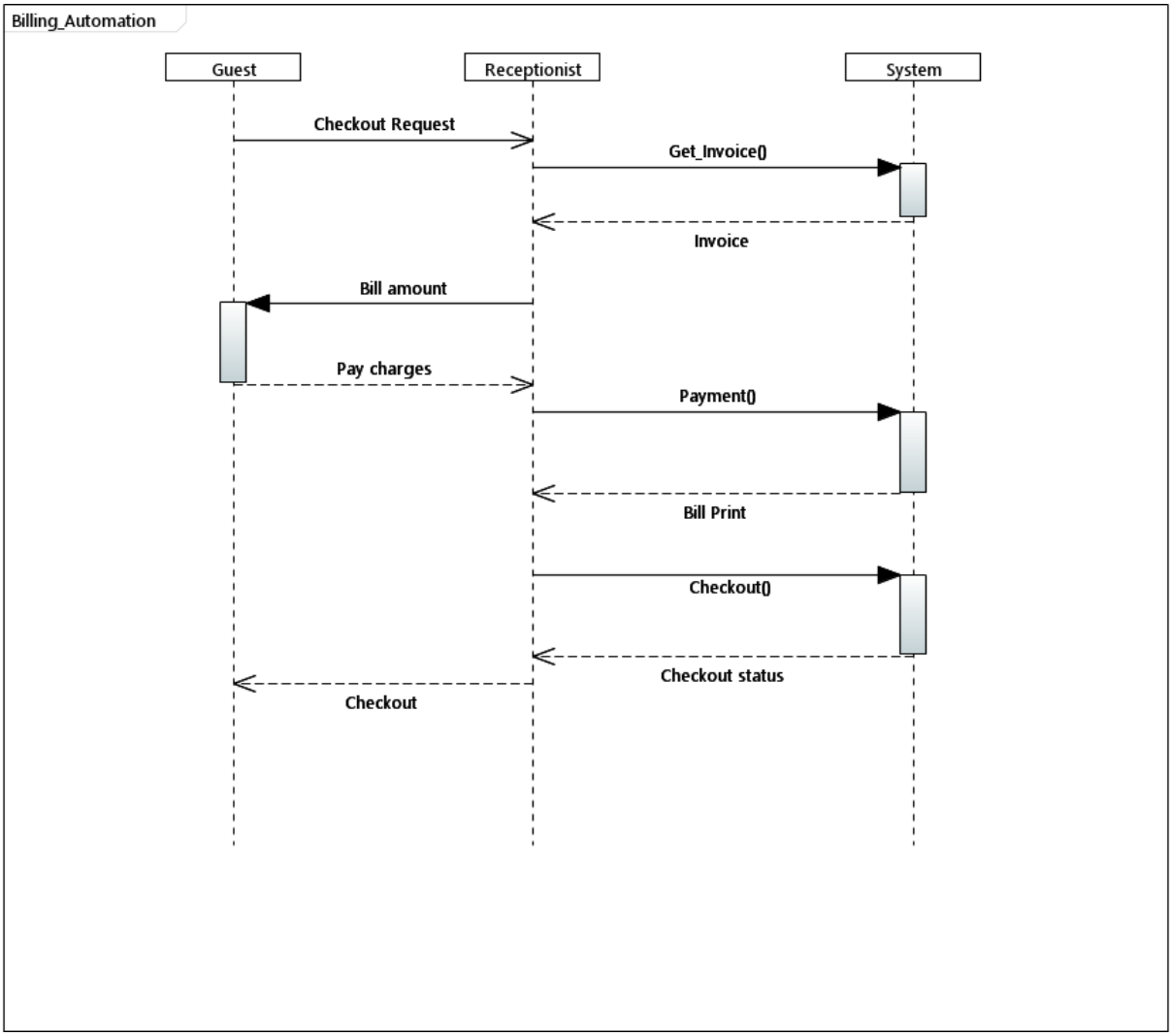
2. Feedback Collection:



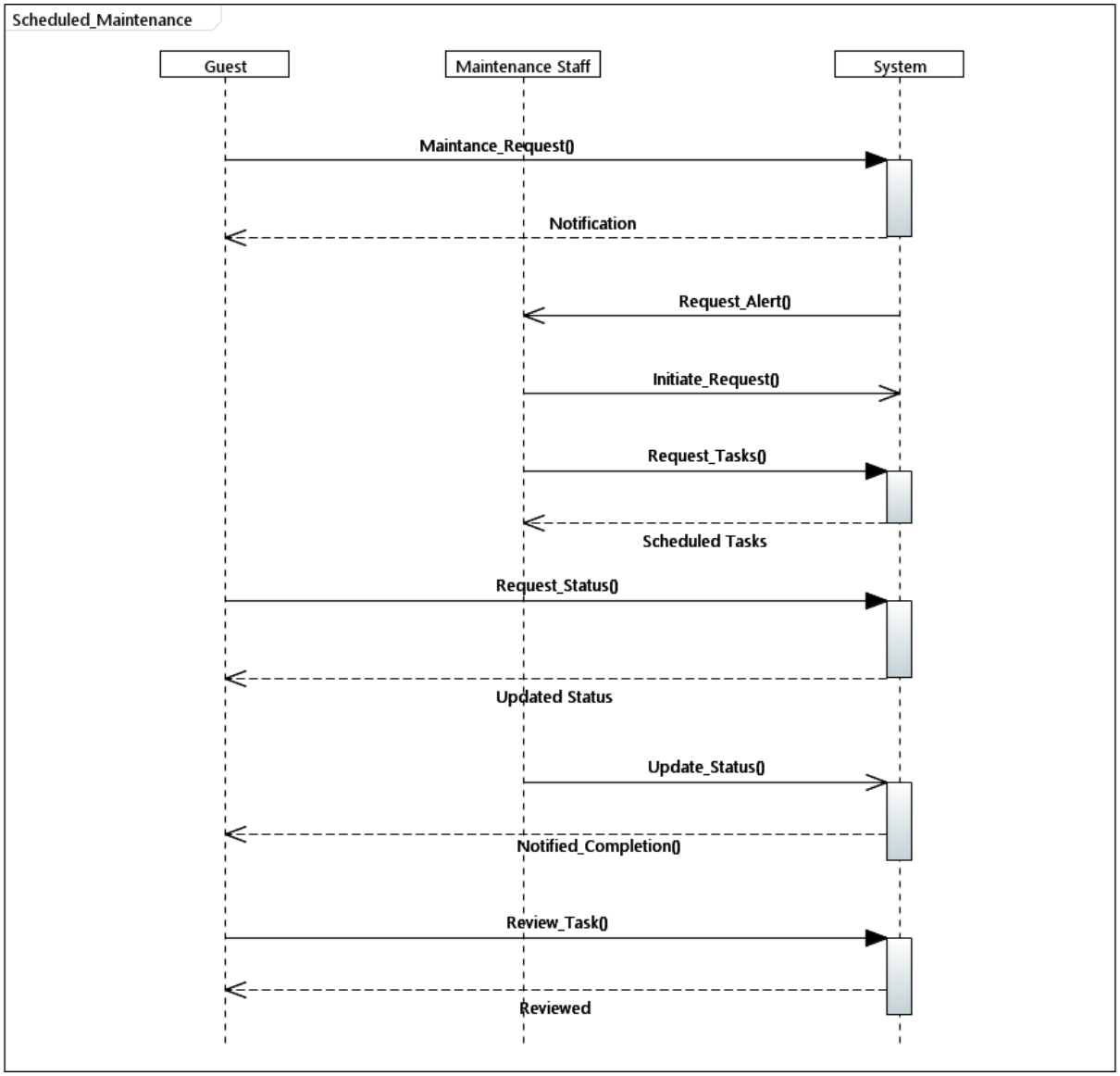
3. Room Maintenance:



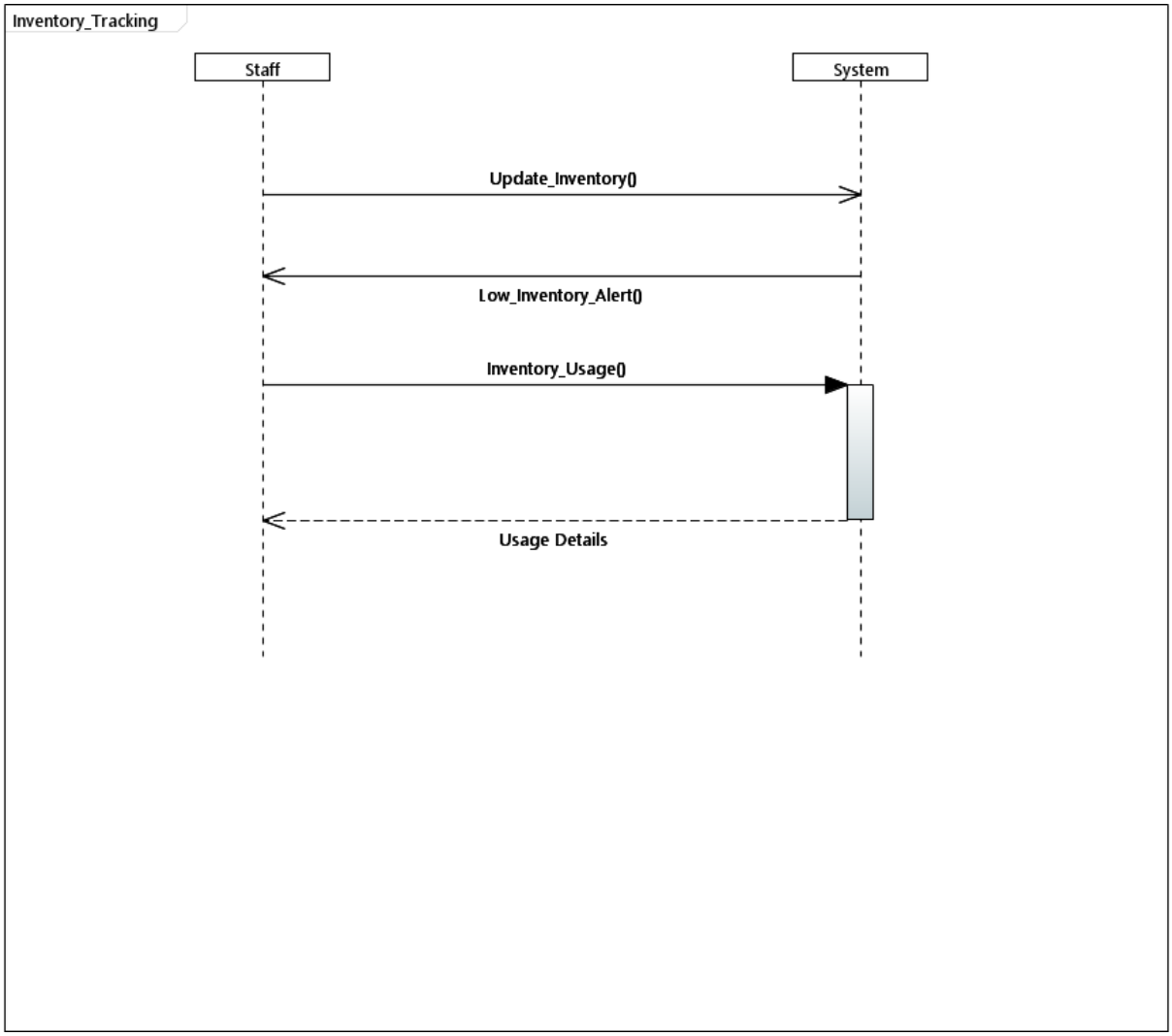
4. Billing Automation:



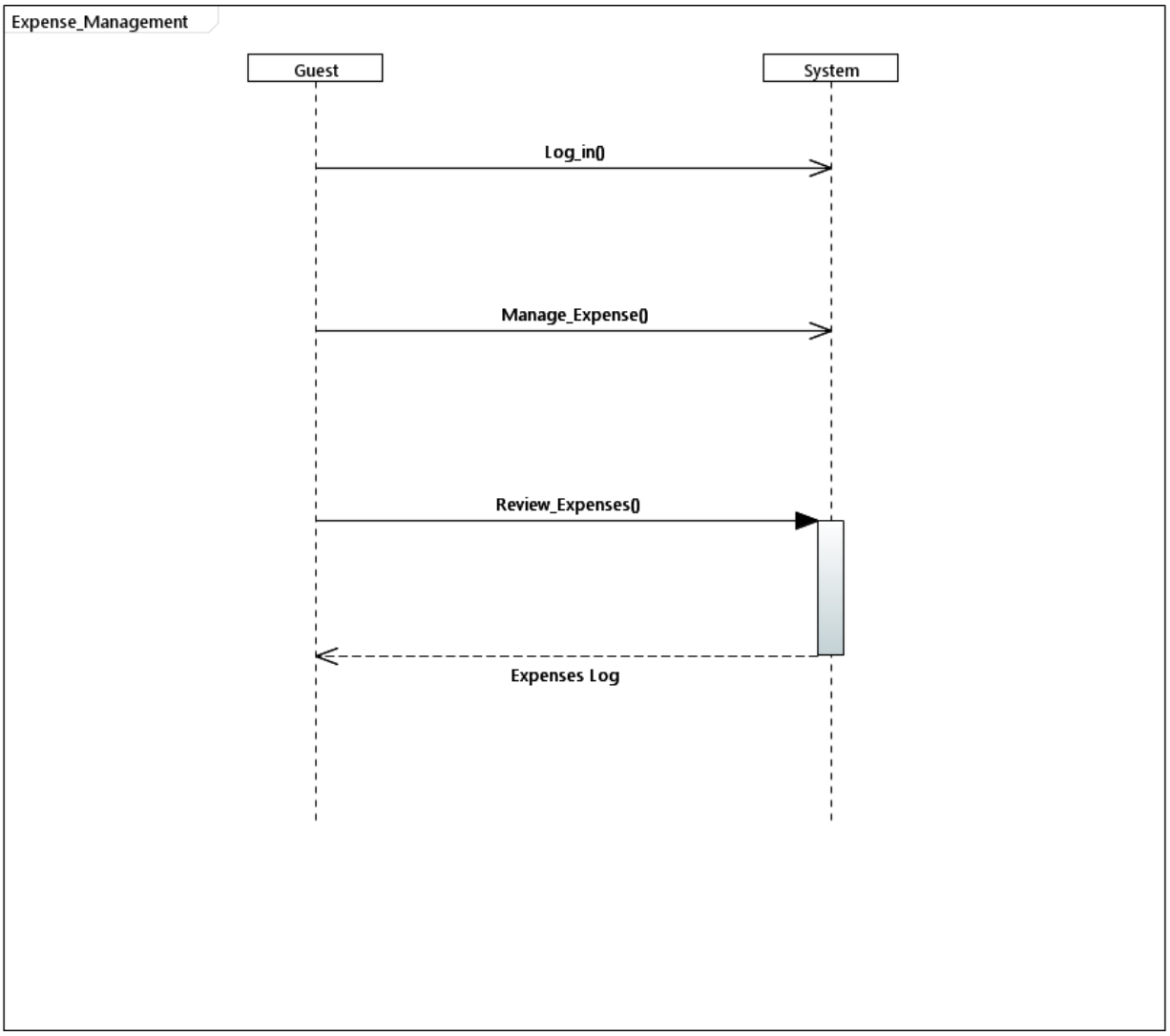
5. Scheduled Maintenance:



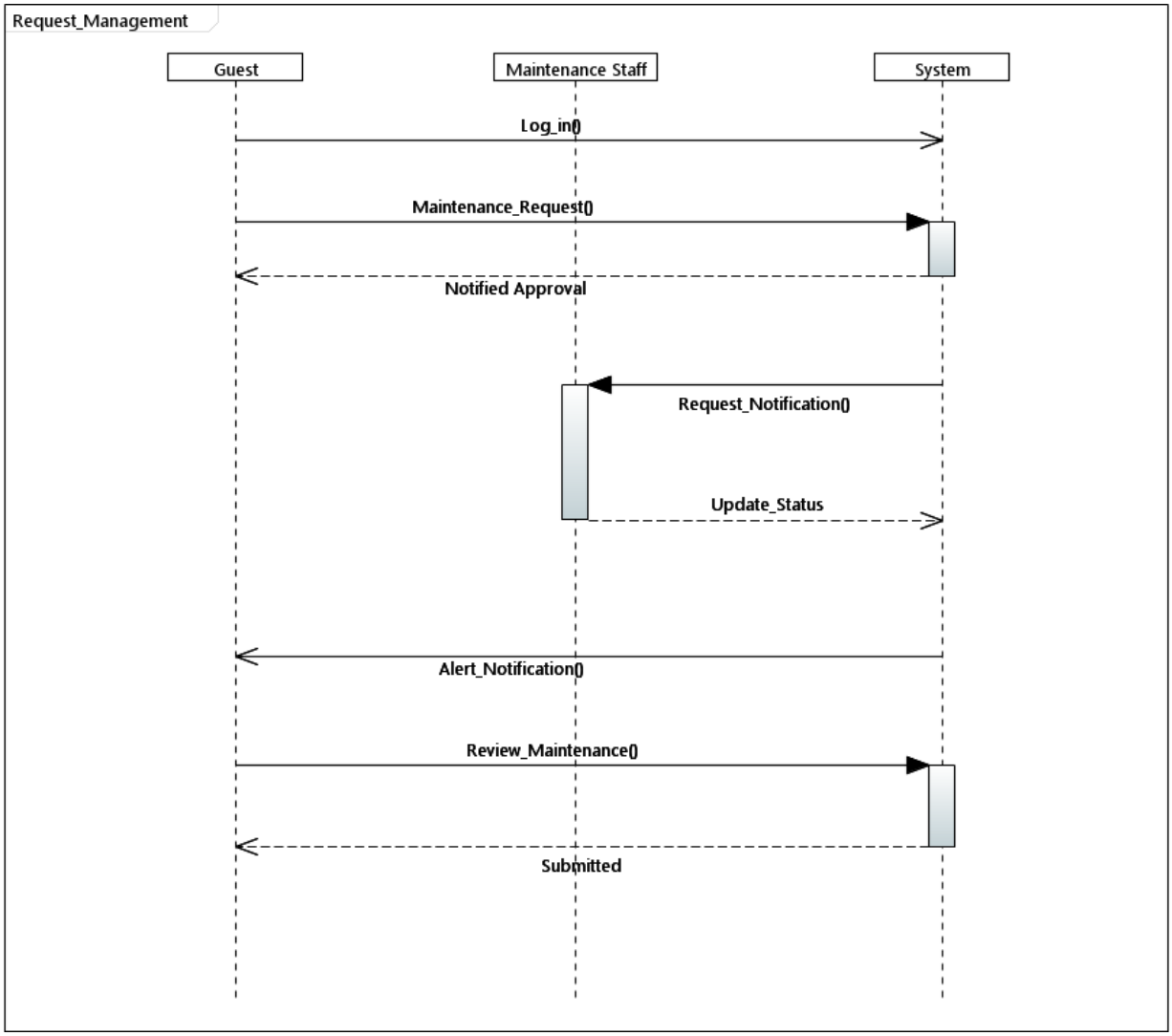
6. Inventory Tracking:



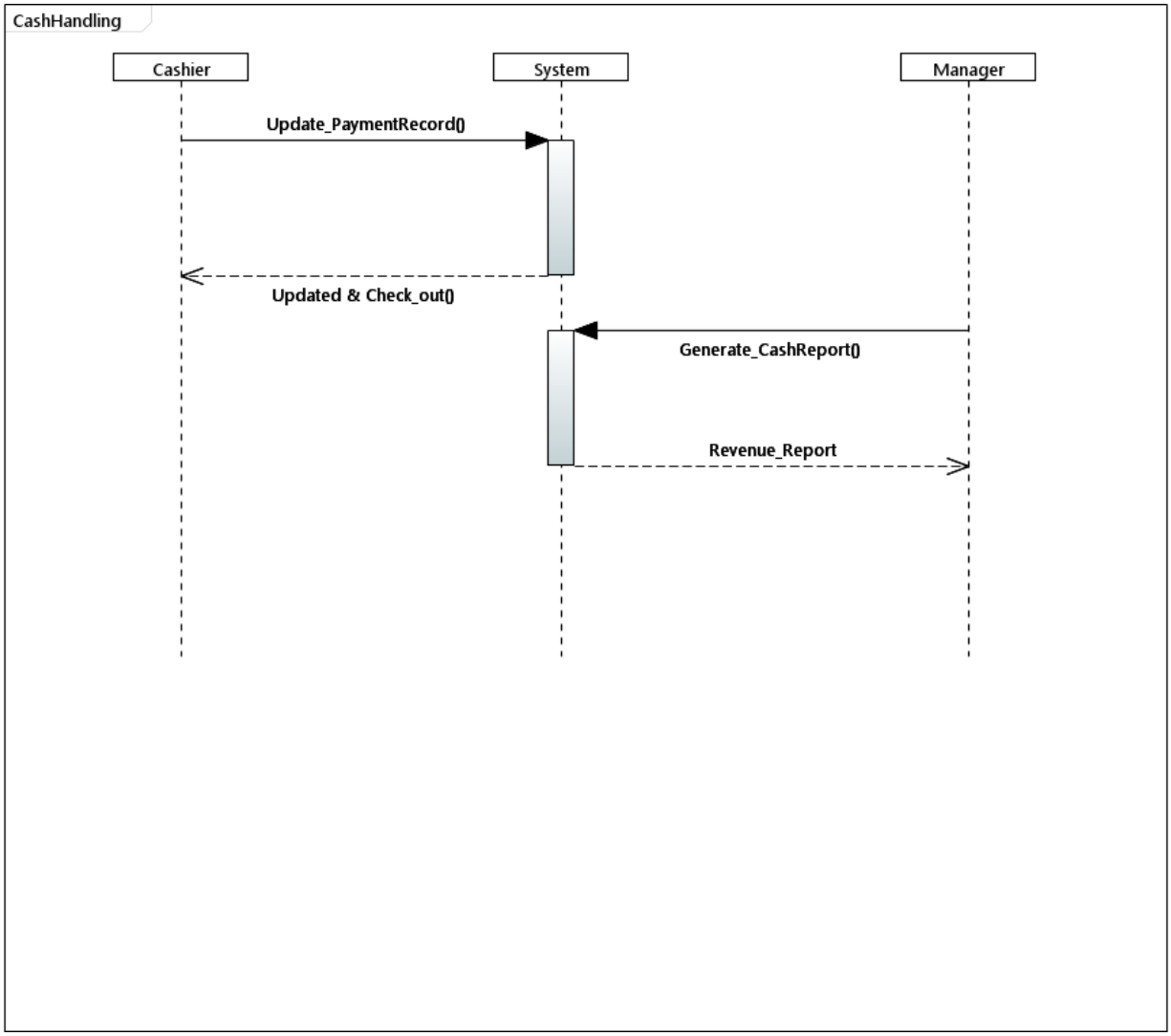
7. Expense Management:



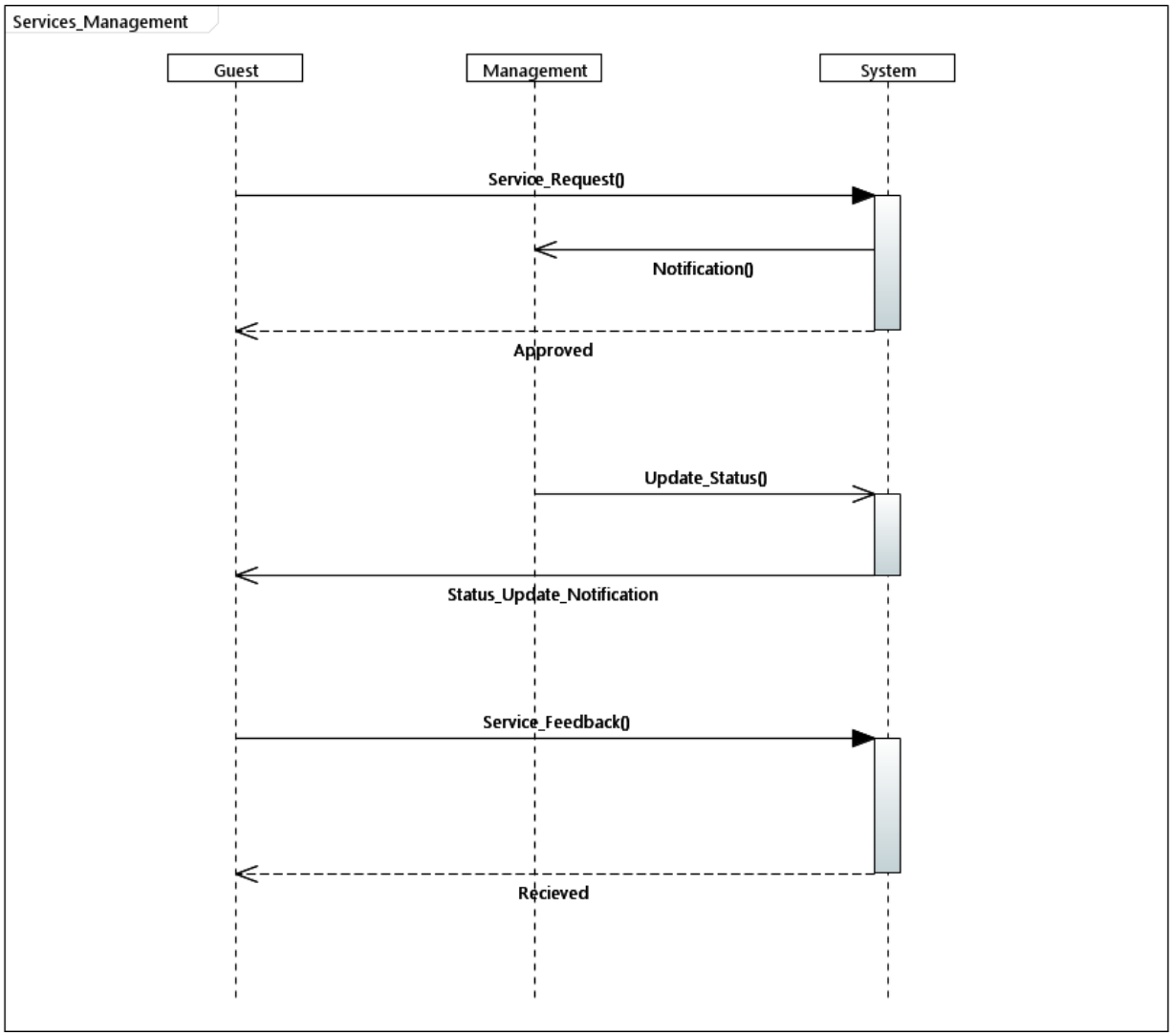
8. Request Management:



9. Cash-Handling:

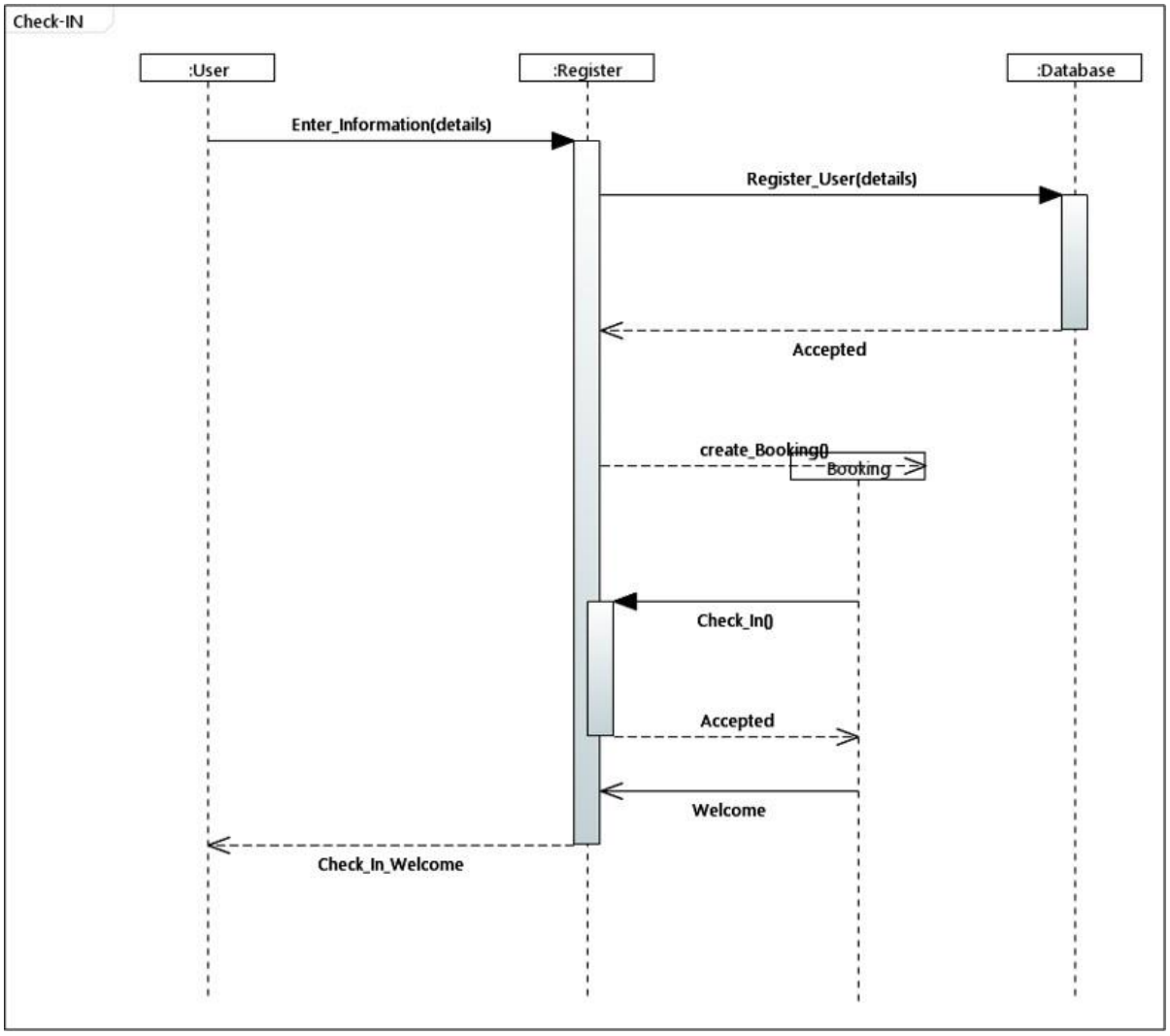


10. Services Management:

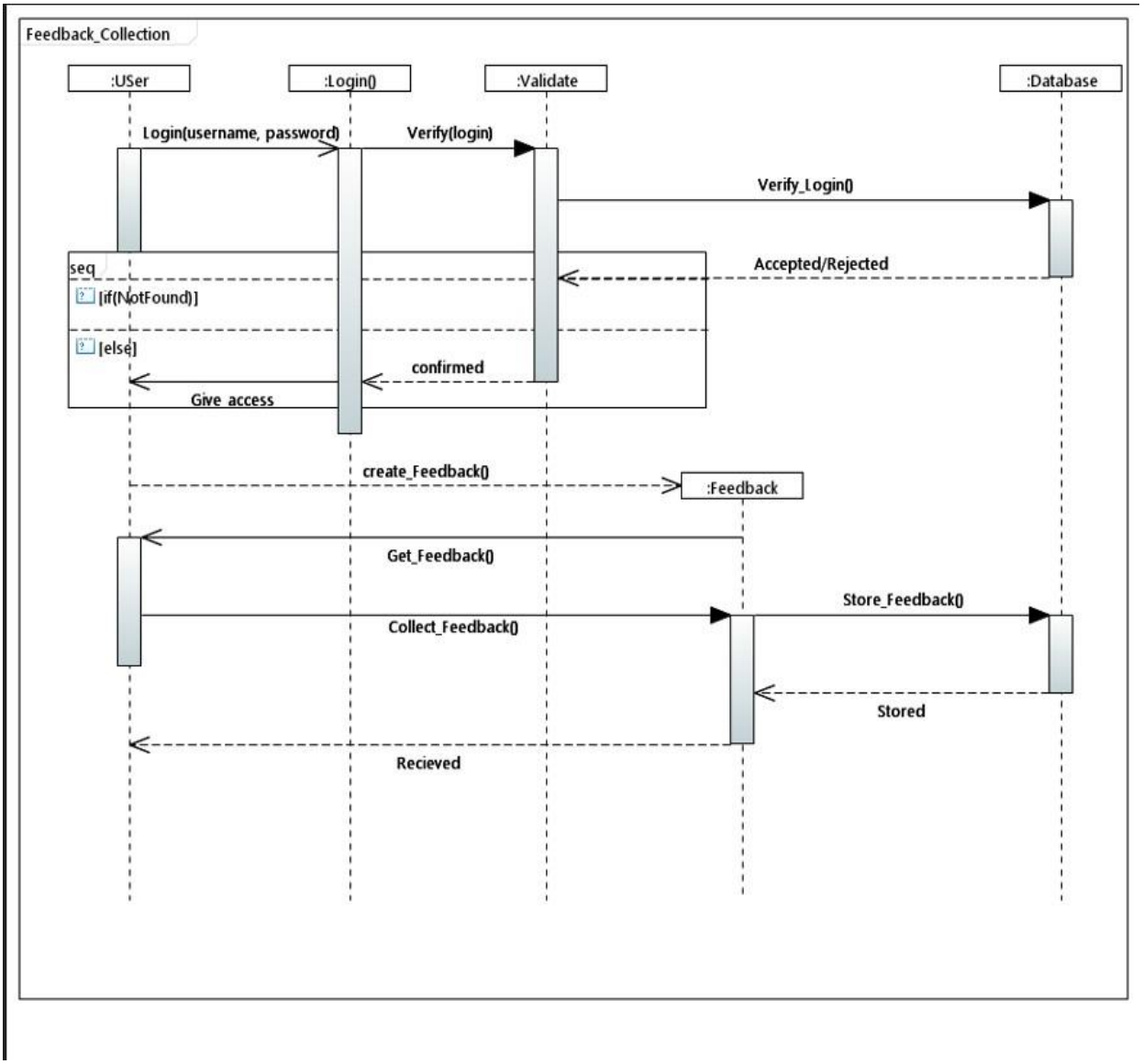


4. Sequence Diagram

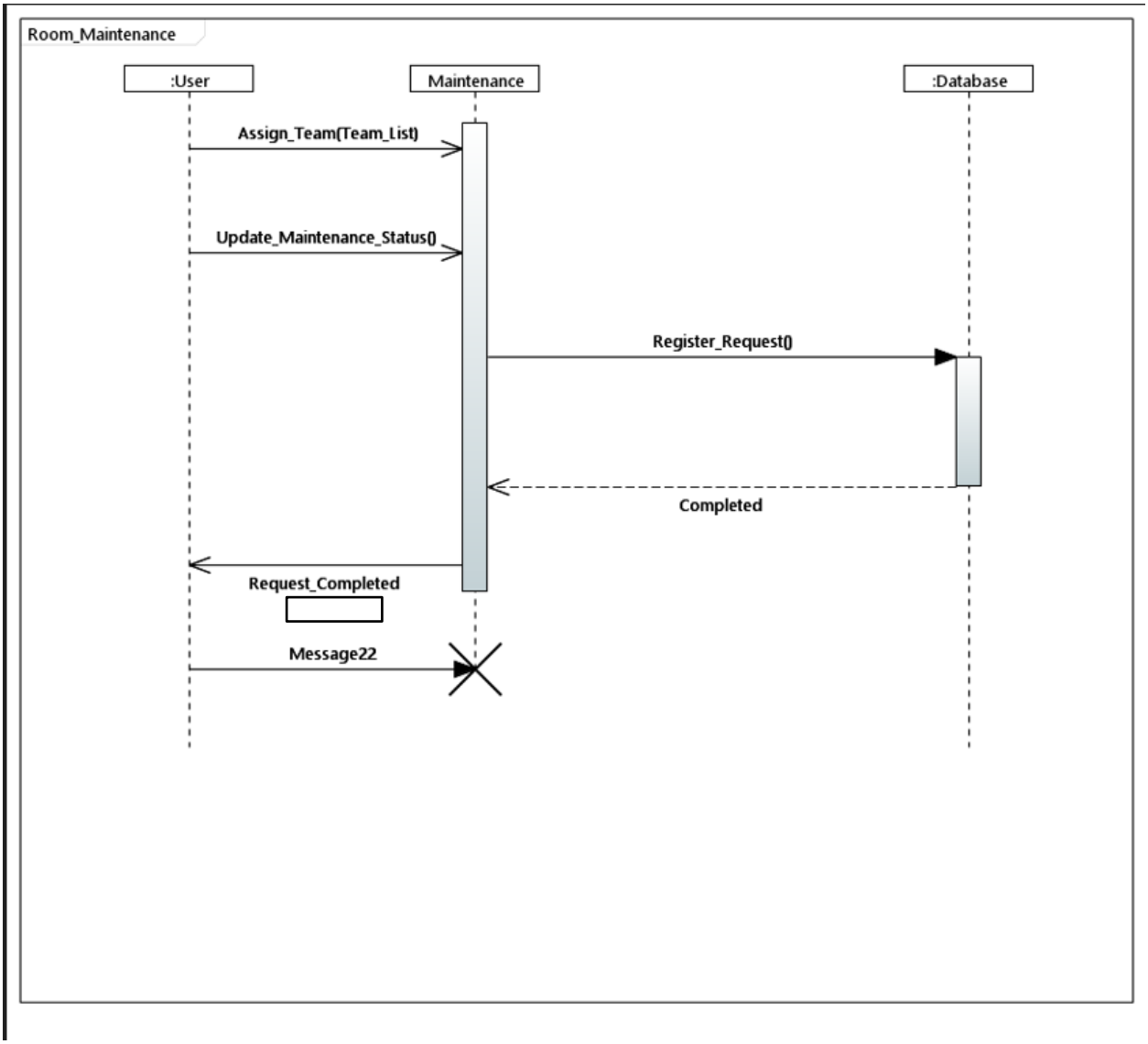
1. Check-In:



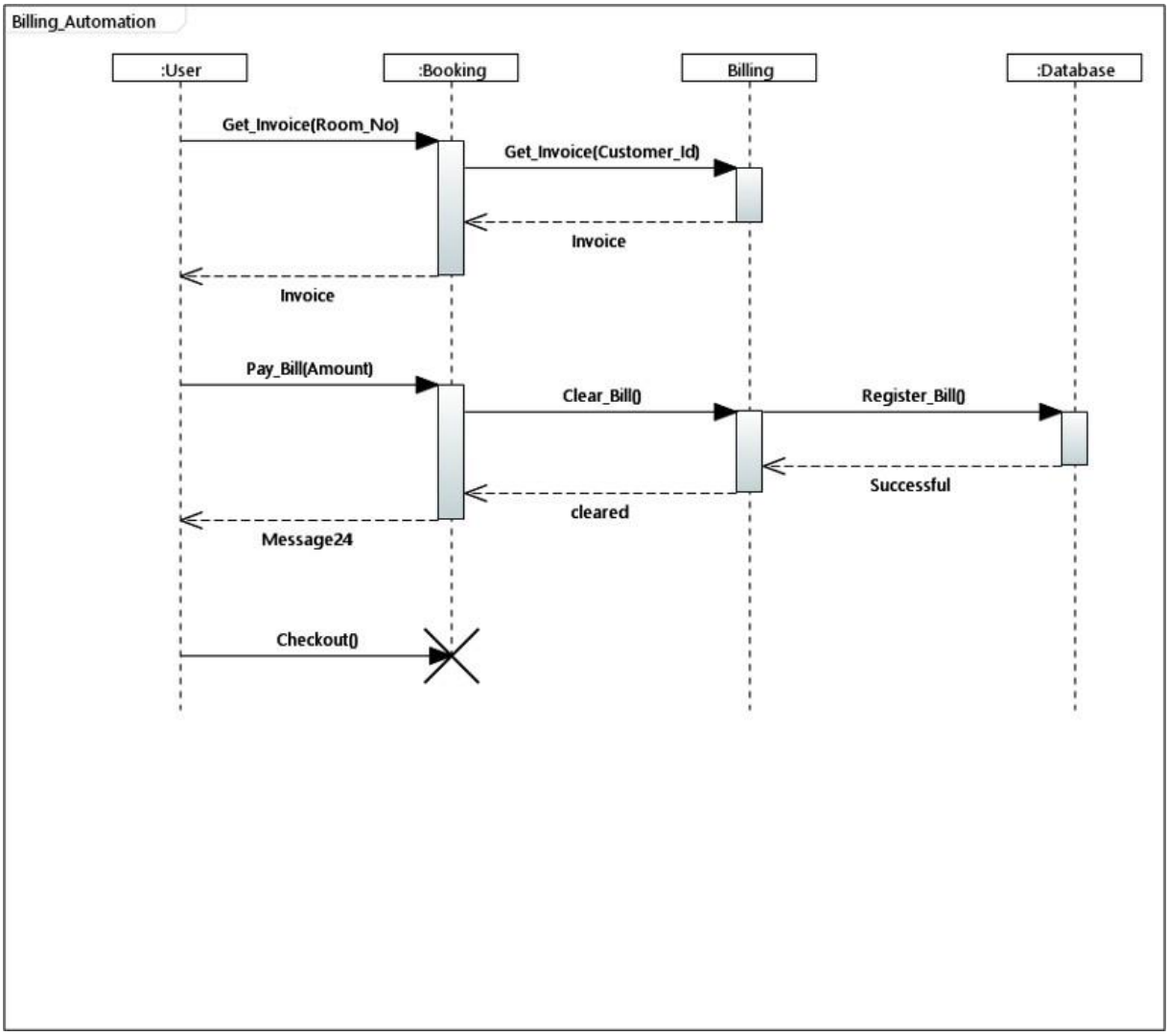
2. Feedback Collection:



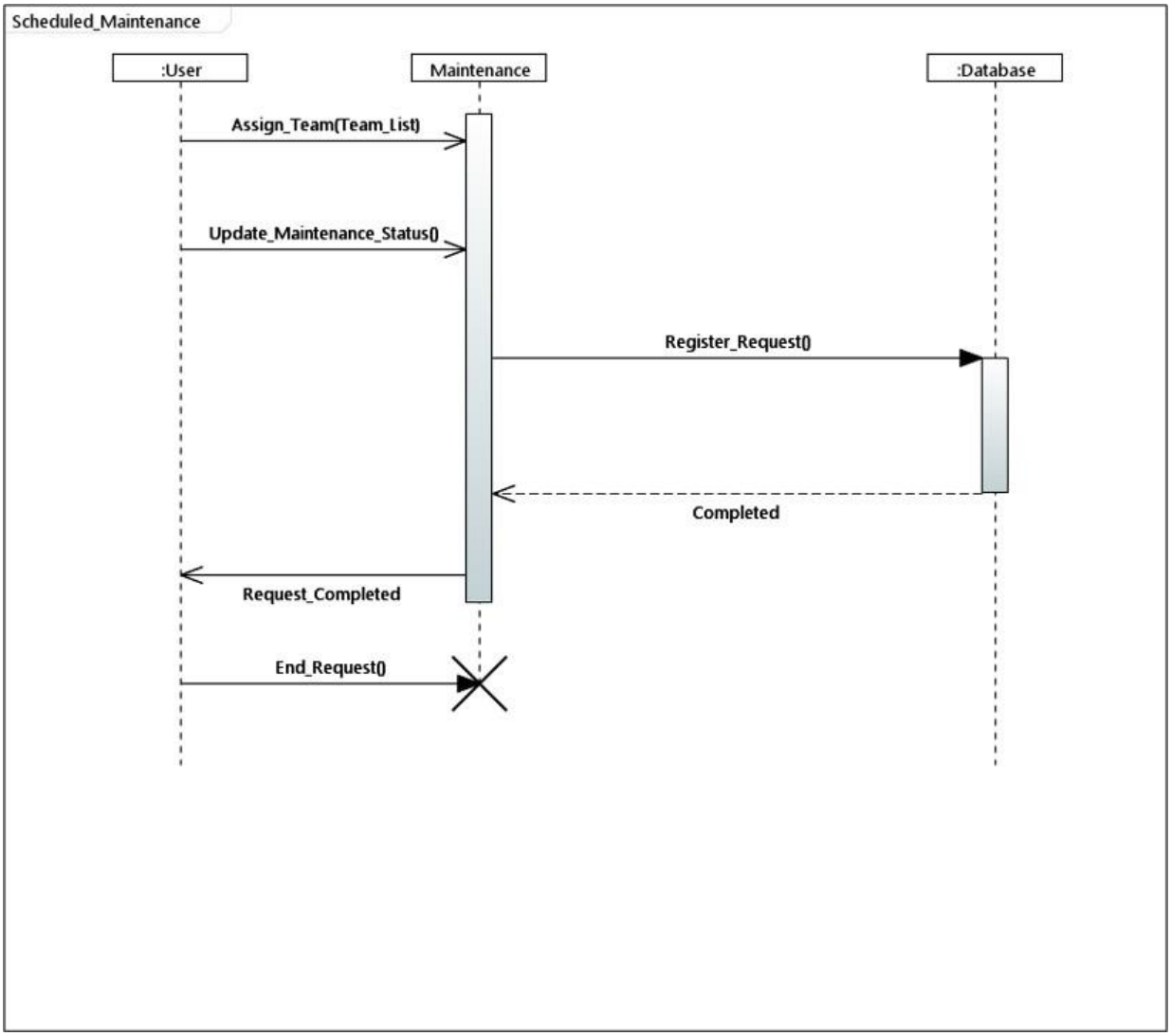
3. Room Maintenance:



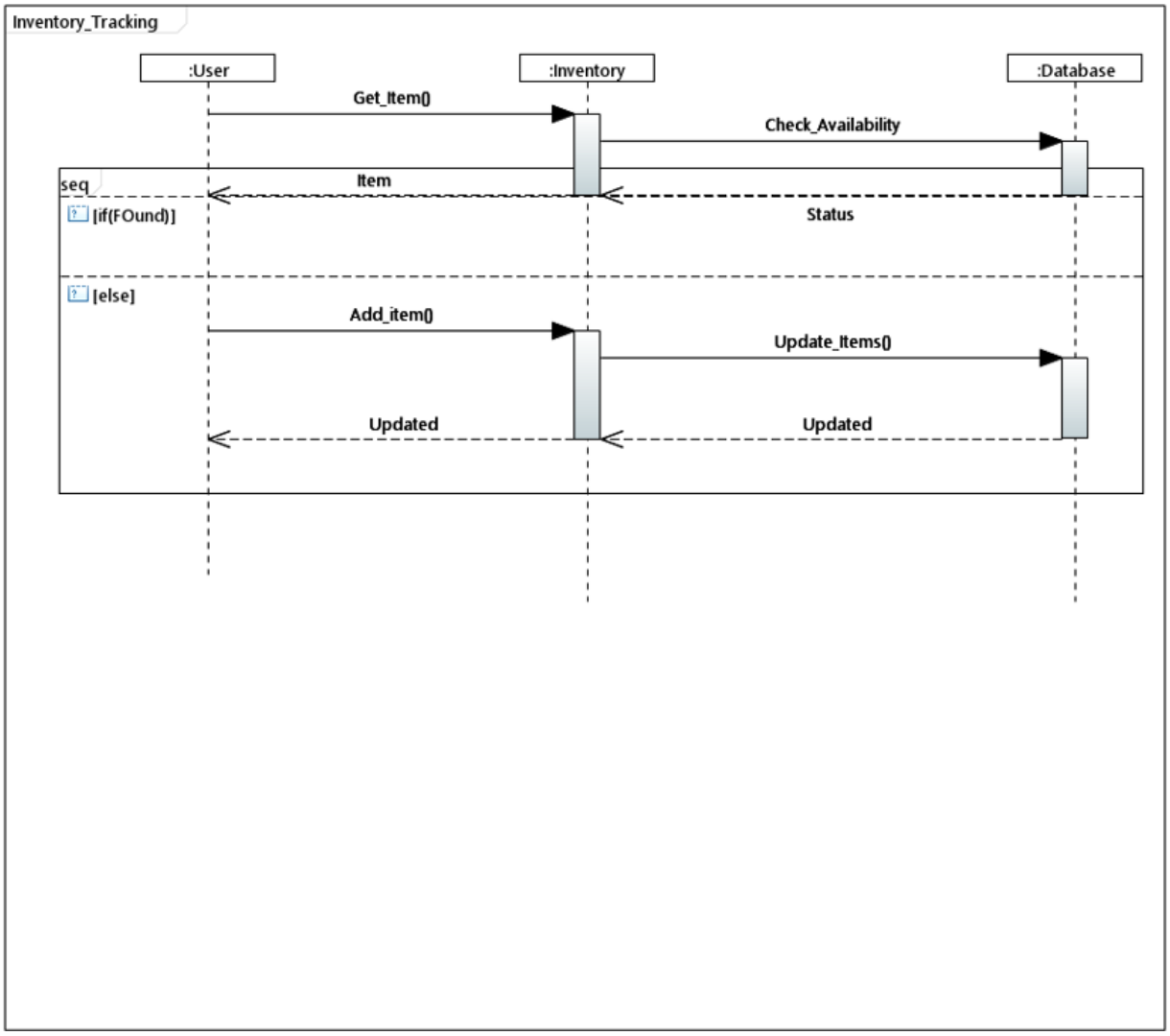
4. Billing Automation:



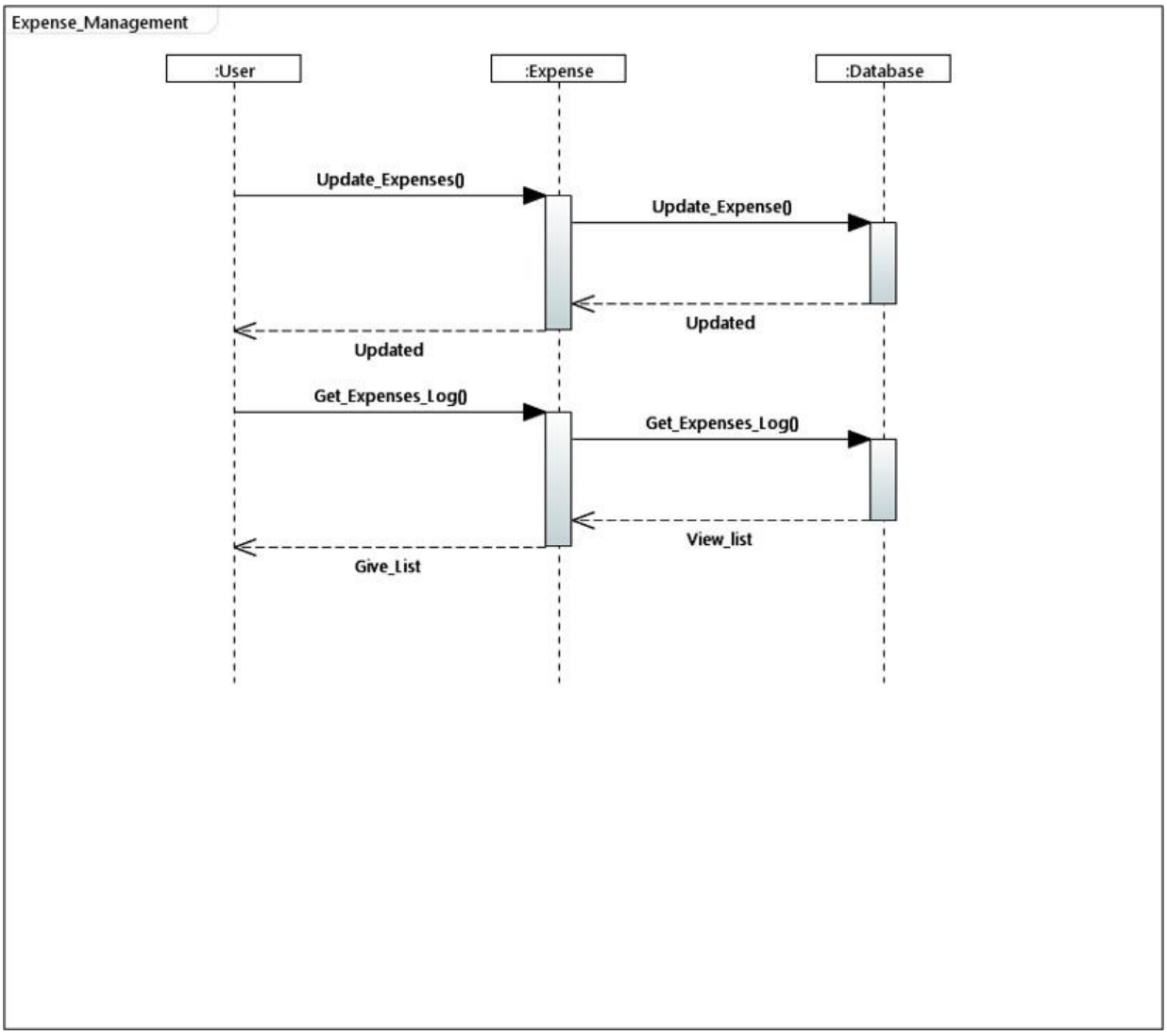
5. Scheduled Maintenance:



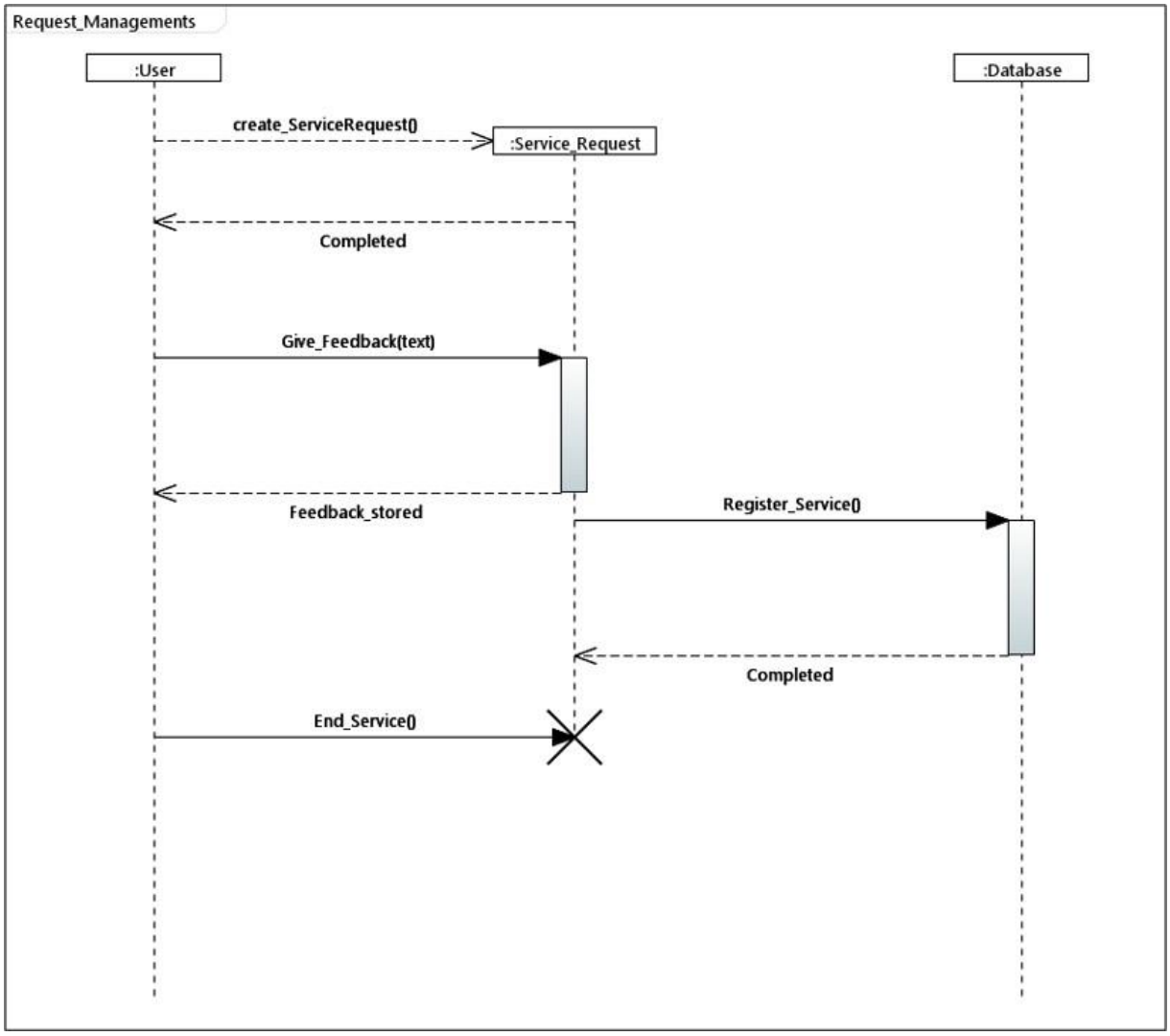
6. Inventory Tracking:



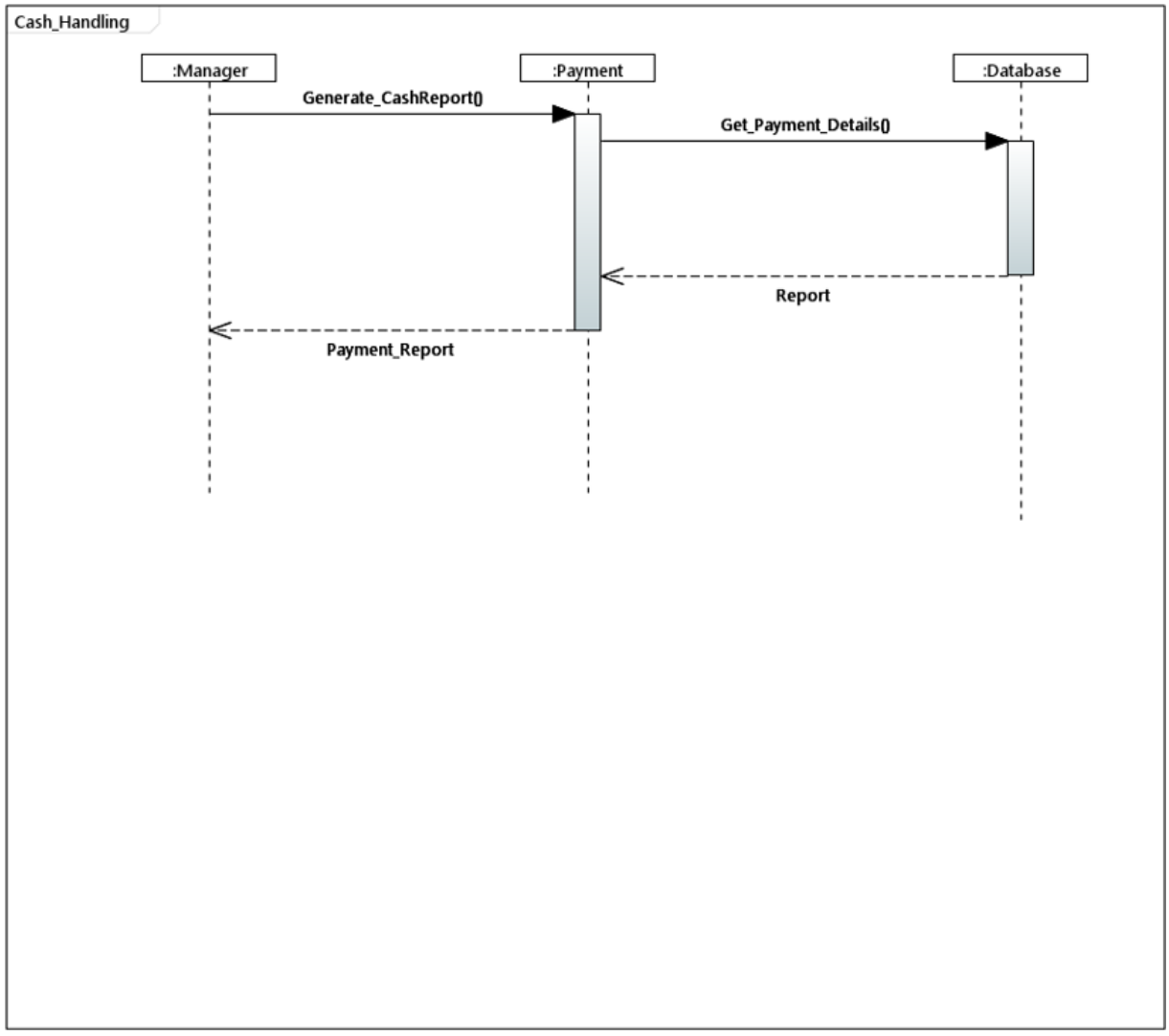
7. Expense Management:



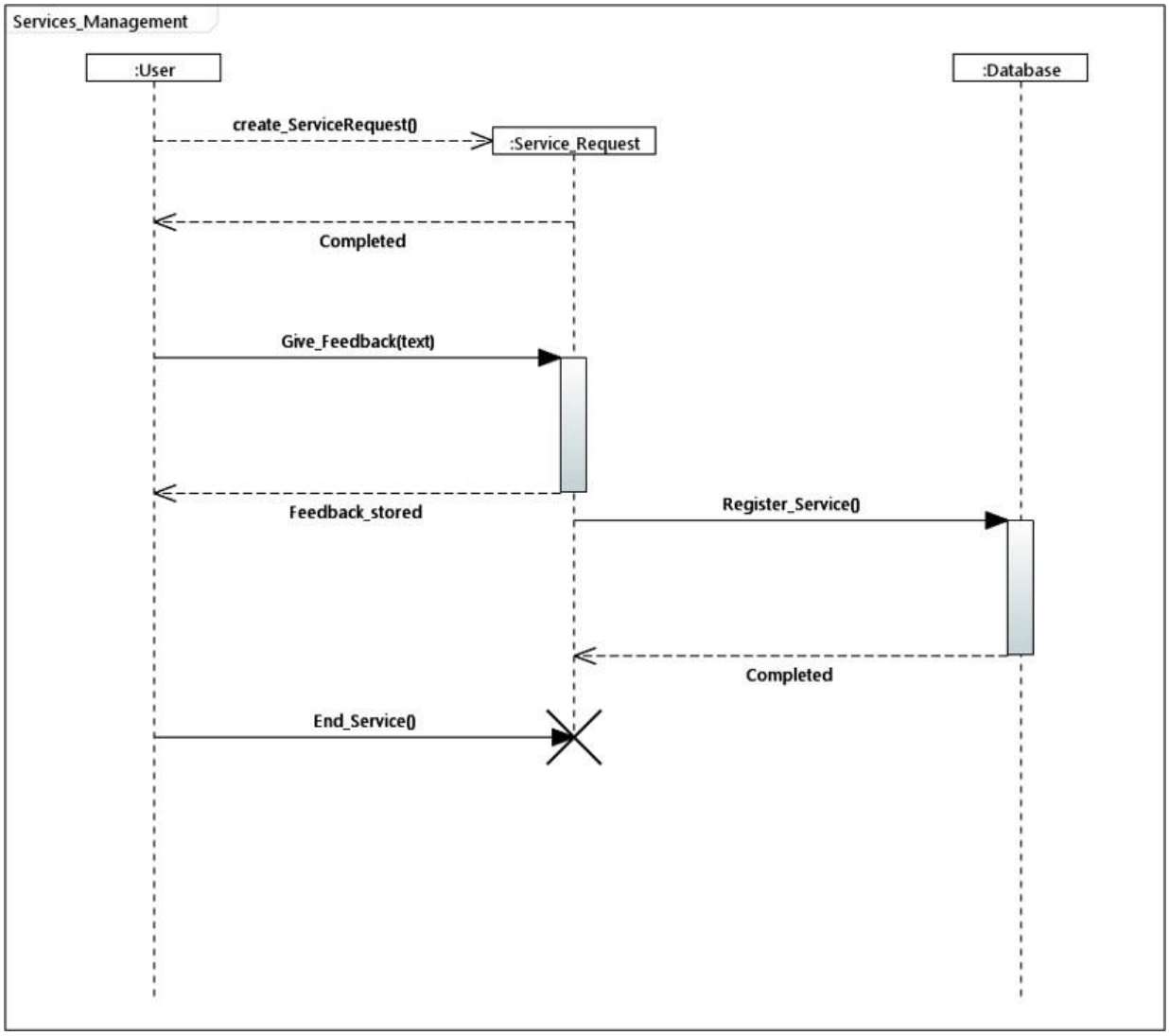
8. Request Management:



9. Cash-Handling:



10. Services Management:



5. Class Diagram