

HOTEL MINI RESERVATION SYSTEM

Presented by:

Shahad Azeb Al-AlHareth, 444301671

Maznah Hadi Al-Rizq,444301987

Shahad Hadi Al-Mutayf, 444305435

Sarah Abdullah Al-Dashel,444302270

Renad Nasser Al-Julaydan,444300506

Supervised by:

Dr. Mohammed Asiri

Agenda

1

Introduction

2

General Description

3

Specific Requirements

4

Analysis Models (UML)

5

Change Management Process

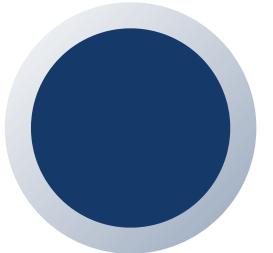
6

Summary

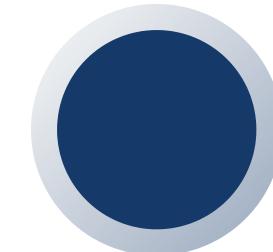
Introduction

The Hotel Mini-Booking System (HRS) is a web-based application designed to manage the core reservation process of a single hotel. It allows customers to create accounts, search for available rooms, make and modify reservations, and complete secure payments using credit card or installment services. Receptionists can retrieve reservation details, verify booking codes, and handle guest check-in and check-out, while the hotel manager can view system-generated monthly reports and overall reservation statistics. The system aims to reduce manual errors, automate payments and invoicing, and improve the efficiency and quality of hotel services through a simple and user-friendly interface.

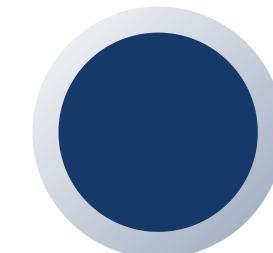
1. Introduction



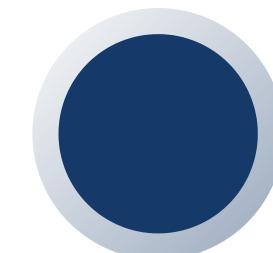
1.1 Purpose



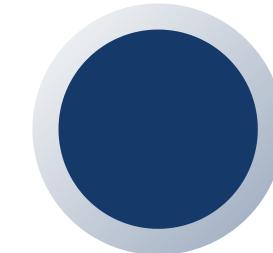
1.2 Scope



**1.3 Definitions,
Acronyms, Abbreviations**



1.4 References



1.5 Overview

Purpose

a) Overview of the Document

This Software Requirements Specification (SRS) describes the complete set of functional and non-functional requirements for the Hotel Mini-Booking System (HRS).

It defines the system behavior, constraints, data requirements, external interactions, and quality attributes.

It also serves as a basis for system design, a reference for developers and testers, and a foundation for acceptance testing.

b) Purpose of the SRS

The purpose of this document is to ensure that all stakeholders share a clear, consistent, and unambiguous understanding of the software system.

This document is prepared as part of the Software Engineering course project at Najran University.

It serves as:

- A basis for system design
- A reference for developers and testers
- A foundation for acceptance testing
- A contract between stakeholders and the development team

c) Intended Audience

This SRS is intended for:

- Software developers and engineers
- System analysts and designers
- Quality assurance and testing teams
- Hotel receptionists and hotel manager
- Project supervisor or instructor
- Maintenance and support teams

Scope

a) Identification of the Software Product:

The software product is the Hotel Mini-Booking System (HRS), a web-based application that supports room reservation, payment processing, check-in/check-out, and reporting.

The HRS interacts with external Payment Systems, such as the Visa Credit Card Gateway and the Installment Payment Service.

b) What the System Will Do:

The HRS will provide the following capabilities:

Customer Functions:

- Create an account and log in securely
- Search available rooms by date and room type
- View room details and facilities
- Make reservations and receive a unique reservation code
- Modify or cancel reservations prior to arrival
- Complete payments through:
 - 1- Credit Card Payment Gateway
 - 2- Installment Payment Service
- Print reservation confirmation
- Receive email notifications
- Extend Reservation

Receptionist Functions

- Retrieve customer reservation details
- Verify reservation codes
- Check guests in
- Check guests out
- Issue invoices

Hotel Manager Functions

- View system-generated monthly reports

c) What the System Will NOT Do

The HRS will not:

- Manage multi-branch hotels
- Support cash payments
- Integrate physical room key systems
- Handle restaurant/spa bookings
- Allow anonymous or walk-in reservations
- Manage payroll or employee scheduling
- Perform full financial accounting

Scope Cont...

d) Application of the Software (Benefits, Objectives, Goals):

Benefits of the HRS:

- Reduces human errors in reservation handling
- Automates payment and invoice generation
- Provides real-time room availability
- Enhances customer satisfaction with faster service
- Reduces receptionist workload
- Ensures secure online transactions through external payment systems

Objectives of the HRS:

- Provide secure online room reservation and payment
- Deliver accurate and traceable reservation records
- Support the complete guest lifecycle (booking → payment → check-in → check-out)
- Integrate safely with credit card and installment payment providers
- Automate reporting for hotel managers

Goals of the HRS:

- Provide a simple, intuitive user interface
- Maintain secure, encrypted operations
- Ensure consistency between reservation and payment modules
- Improve hotel operational efficiency

Definitions, Acronyms, Abbreviation

a) Definitions:

Term	Definition
Customer	A user who browses rooms and creates reservations.
Receptionist	Staff member who performs check-in/check-out and reservation management.
Hotel Manager	Administrative user who views monthly system reports.
Reservation	A confirmed booking including dates, room type, and guest details.
Reservation Code	A unique identifier assigned to each reservation.
Check-in	Process that verifies a guest's reservation and marks them as arrived.
Check-out	Completion of stay including payment and invoice issuance.
Availability	Indicates whether a room is free for selected dates.
Invoice	Document issued after payment at check-out.

b) Acronyms:

Acronym	Meaning
SRS	Software Requirements Specification
HRS	Hotel Reservation System
API	Application Programming Interface
DB	Database
UI	User Interface

c) Abbreviations

Abbreviation	Meaning
Cust.	Customer
Rec.	Receptionist
Res.	Reservation
Mgr.	Manager

References

- The following references were used in preparing this Software Requirements Specification (SRS):
- [1] Bruegge, B., & Dutoit, A. H. (2009). Object-Oriented Software Engineering: Using UML, Patterns, and Java (3rd ed.). Prentice-Hall.
 - [2] Fowler, M., & Scott, K. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley.
 - [3] Visual Paradigm. “What is a Sequence Diagram?” Available:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
 - [4] GeeksforGeeks. “Software Requirement Specification (SRS) Format.” Available:
<https://www.geeksforgeeks.org/software-engineering/software-requirement-specification-srs-format/>
 - [5] DSPMU Ranchi. “SRS Template – IEEE Format.” Available:
https://dspmuranchi.ac.in/pdf/Blog/srs_template-ieee.pdf
 - [6] Modern Requirements. “How to Write a Software Requirements Specification (SRS) Document.” Available:
<https://www.modernrequirements.com/blogs/requirements-specification/>
 - [7] Perforce Software. “How to Write an SRS Document.” Available:
<https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
 - [8] GeeksforGeeks. “UML Activity Diagrams.” Available:
<https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-activity-diagrams/>

Overview

The remainder of this Project contains:

- **Section No.2 General Description:**

Provides background on system users, product perspective, constraints, assumptions, and the operational environment.

- **Section No.3 Specific Requirements:**

Lists all functional requirements (FR-01 to FR-13), non-functional requirements, data models, external interfaces, and the traceability matrix mapping each FR to its related UML diagrams.

- **Section No.4 Analysis Models:**

Includes the full set of UML diagrams describing the structural and behavioral aspects of the system.

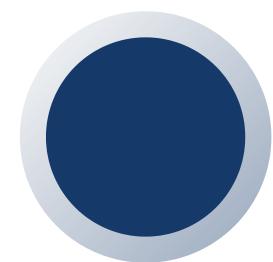
- **Section No.5 Change Management Process:**

Describes how requirement changes are requested, reviewed, approved, and versioned.

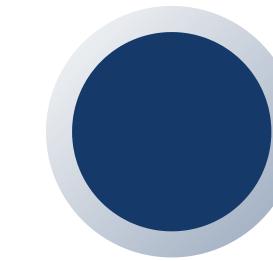
- **Appendices:**

Contain glossary terms, supporting notes, API samples, and UI drafts.

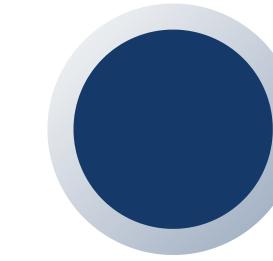
2. General Description



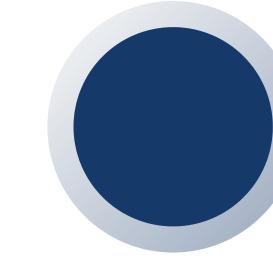
2.1 Product Perspective



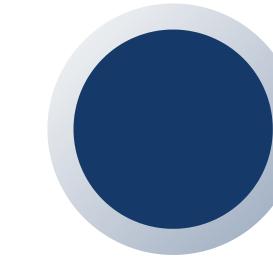
2.2 Product Functions



2.3 User Characteristics



2.4 General Constraints



2.5 Assumptions and Dependencies

Product Perspective

The Hotel Mini-Booking System (HRS) is a web-based application that supports room reservations, payment processing, and guest check-in/check-out.

It operates as a standalone reservation system while interacting with several external components:

- Room Database – stores room types, room numbers, and availability data.
- User Database – stores customer profiles, login credentials, and contact information.
- External Payment Services – such as the Visa Credit Card Gateway and the Installment Payment Service.

The system provides different user interfaces based on user roles:

- Customer interface – a simplified UI for searching rooms, viewing details, making reservations, completing payments, printing confirmations, and receiving notifications.
- Receptionist interface – an internal UI for managing reservations, retrieving customer details, verifying reservation codes, and performing check-in and check-out.
- Hotel manager interface – a reporting UI that allows viewing monthly reports and high-level reservation statistics.

Each user role has clearly defined permissions and access levels to keep the system secure, organized, and easy to use.

Product Functions

The core functions of the HRS can be summarized as follows:

-For Customers

- Create a new user account.
- Log in using secure credentials.
- Search and select available rooms based on dates and room type.
- View detailed room information and facilities.
- Make reservations and receive a unique reservation code.
- Modify or cancel reservations prior to arrival.
- Process payments using:
 - Credit card payment gateway
 - Installment payment service
- View and print reservation confirmations.
- Receive email notifications (confirmation, reminders, cancellation alerts, etc.).

-For Receptionists

- Retrieve customer reservation details using reservation code or customer data.
- Edit, update, or cancel reservations upon request.
- Validate reservations and perform guest check-in and check-out.
- Issue invoices and confirm final payment at check-out.

-For Hotel Manager

- View system-generated monthly reports and reservation statistics.
- Monitor overall performance of the reservation and payment processes.



User Characteristics

The system targets three main user groups:

1.

Customers:

- **Are not required to have technical expertise.**
- **Should be able to navigate a simple, user-friendly web interface.**
- **Are expected to provide valid personal and payment information.**

2.

Receptionists (Hotel Staff):

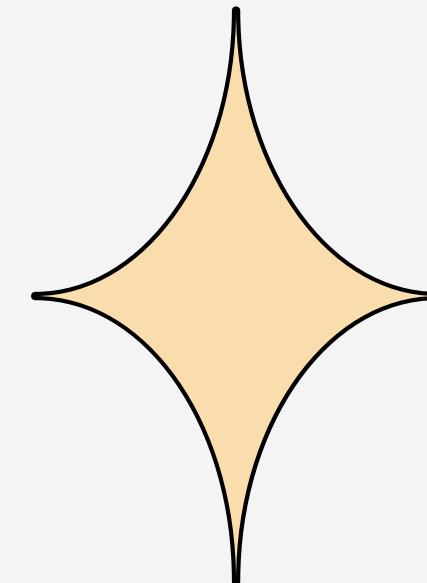
- **Are familiar with hotel operational procedures.**
- **Should be trained to manage reservations and handle check-in/check-out.**
- **Require accurate and real-time access to reservation and payment information.**

3.

Hotel Manager:

- **Requires high-level reports and aggregated reservation statistics.**
- **Is expected to interpret system-generated financial and operational summaries for decision-making.**

General Constraints



- A stable internet connection is required for normal system operation.
- Sensitive payment information must be handled securely and in accordance with hotel privacy and data protection policies.
- External payment services (Visa credit card gateway and Installment Payment Service) must be available for payment transactions to succeed.
- Room availability must be synchronized in real time with the Room Database.
- Only authorized staff members can access administrative and reporting functions.
- The system should support modern, standards-compliant web browsers.

Assumptions and Dependencies

-Assumptions:

- Customers provide correct and up-to-date personal and payment details.
- External payment gateways (Visa, Installments) are operational at the time of the transaction.
- Receptionists use the system correctly to update and manage reservations.
- All users have access to the internet and a compatible device (PC, laptop, or mobile).

-Dependencies:

The HRS depends on the following components:

- Room Database – for real-time room availability and room details.
- User Database – for user authentication and customer profiles.
- Payment Manager and external payment services (Visa Credit Card Gateway and Installment Payment Service) – for processing payments and confirming transactions.
- Network infrastructure – for communication between the HRS, databases, and external payment services.

3. Specific Requirements

- 3.1 External Interface Requirements:**
- 3.2 Functional Requirements — use “shall” statements with IDs (FR-XX).**
- 3.3 Non-Functional Requirements — measurable targets with IDs (NFR-XX).**
- 3.4 Data / Logical Model — key entities/fields.**
- 3.5 Use Cases — brief texts; reference diagram in Section 4.**
- 3.6 Acceptance Criteria — scenario → measurable expected result.**
- 3.7 Traceability Matrix — FR → UC → Sequence → Class → Activity → State.**

EXTERNAL INTERFACE REQUIREMENTS

USER INTERFACE:

Flow: Login/Register → Browse Rooms → Select Room → Choose Payment (Card/Installment) → Print Invoice;

accessible text (Arial ≥ 18pt); buttons ≥ 44px; clear error messages.

HARDWARE:

Client: Touchscreen Mobile/PC;

Hotel: Receptionist Workstation; Printer: Standard Office Printer.

EXTERNAL INTERFACE REQUIREMENTS CONT.

SOFTWARE/APIS:

Endpoints: GET /rooms/availability, POST /auth/login, POST /payments (Card/Installment), GET /reports;

Integrations: Visa Service, Installment Service;

JSON format for errors.

COMMUNICATIONS:

HTTPS/TLS 1.2+ (Secure Transactions);
SMTP (for email notifications);
NTP (for timestamp sync).

Functional Requirements

FR-ID	Requirement (shall)	Notes / Source
FR-01	The system shall allow users to Log in using email/password.	Security / System
FR-02	The system shall allow Customers to Create an Account.	UC: Create Account
FR-03	The system shall allow Customers to Search for available rooms by date.	UC: Specify Room Type & Dates
FR-04	The system shall allow Customers to Make a Reservation.	UC: Make Reservation
FR-05	The system shall allow Customers and Receptionists to Edit/Cancel, or Extend Reservations.	UC: Edit Reservation, UC: Cancel Reservation
FR-06	The system shall retrieve Customer Details for the Receptionist.	UC: Retrieve Customer Details
FR-07	The system shall allow Receptionists to Check-in Guests.	UC: Check-in Guest
FR-08	The system shall allow Receptionists to Check-out Guests.	UC: Check-out Guest
FR-09	The system shall process payments via Credit Card or Installments.	UC: Process Payment
FR-10	The system shall issue and print Invoices for guests.	UC: Issue Invoice
FR-11	The system shall send Notifications to Customers.	UC: Send Notifications
FR-12	The system shall generate Monthly Reports for the Manager.	UC: View Monthly Reports
FR-13	The system shall encrypt payment data for security.	Security / Payment

Data - Logical Model

- **Customer** (customerID, name, email, contactNumber)
- **Reservation** (reservationID, reservationCode, checkInDate, checkOutDate, numberOfGuests, status)
- **Room** (roomNumber, status, type)
- **RoomType** (roomTypeID, typeName, maxGuests, facilities)
- **Invoice** (invoiceID, invoiceDate, totalAmount)
- **Payment** (paymentID, amount, paymentDate, status) **[base entity]**
- **CreditCard** (cardNumber, cardHolderName, expiryDate) **[payment extension]**
- **Installment** (installmentPlanID, installmentCount) **[payment extension]**
- **MonthlyReport** (reportID, month, year, totalReservations)
- **Staff** (staffID, fname, lname) **[includes Receptionist & HotelManager]**

Non-Functional Requirements

Non-functional requirements describe how the system should perform rather than what it should do. They define the system's quality attributes such as performance, usability, security, reliability, and maintainability.

performance

usability

security

maintainability.

reliability

Compatibility

Scalability

Performance Requirements (FR-08)

Requirement	Details
Performance	<ul style="list-style-type: none">The system must load all pages within 3 seconds.Database queries must respond within 2 seconds.
Description	<ul style="list-style-type: none">The system must load all pages within 3 seconds.

Usability Requirements (FR-07)

Usability	<ul style="list-style-type: none">The interface must be simple and easy to navigate.Buttons and forms must be clear and intuitive.Supports mobile and desktop.
Security	<ul style="list-style-type: none">Passwords must be encrypted.User data must be protected.All transactions must use HTTPS.

Non-Functional Requirement: Reliability (NFR-04)

Reliability	<ul style="list-style-type: none">System uptime must be 99%.Automatic recovery after crashes.
Compatibility Requirements	<ul style="list-style-type: none">Automatic recovery after crashes.

Scalability Requirements (NFR-05)	
Scalability	<ul style="list-style-type: none"> System supports increasing users without slowing down. Database handles large reservation data smoothly. Supports mobile and desktop.

Compatibility Requirement (NFR-06)	
Requirement	Works on Chrome, Safari, Edge, Firefox.
Details	Runs on Windows, macOS, iOS, Android.
Scalability Requirements	System supports increasing users without slowing down.
Compatibility Requirements	Works on Chrome, Safari, Edge, Firefox.

Maintainability Requirements	
Category	Requirement
Maintainability	<ul style="list-style-type: none"> Code must be clean and easy to update. Future updates should not break existing features. Supports mobile and desktop.

(Use Case Descriptions)

The following use-case tables provide a clear and simple description of the main system functions.

Each table summarizes the primary actor, the goal, the main steps, and the expected outcome.

These use cases help understand how the system should behave and what each user can do.

Use Case: Create Account (FR-02)

Primary Actor	Customer
Goal	Register a new account.
Pre-condition	User does nt have an existing account.
Main Flow	<ol style="list-style-type: none"> 1. Customer opens the system and selects "Create Account." 2. System validates the information. 3. System creates a new customer profile. 5. System sends confirmation to the user.
Post-condition	A new customer account exists in the system.
Post-condition	A new customer account exists in the system.

Use Case: Specify Room Type & Dates (FR-03)

Primary Actor	Customer
Goal	Search for available rooms by type and date.
Pre-condition	User is logged in.
Main Flow	<ol style="list-style-type: none"> 1. Customer selects room type and dates. 2. System returns list of available rooms. 3. System displays availability status.
Post-condition	List of available rooms has been returned.
Post-condition	A new available rooms has been returned.

Use Case: Make Reservation (FR-04)

Primary Actor	Customer
Goal	Reserve a new account.
Pre-condition	User does not have an existing account.
Main Flow	<ol style="list-style-type: none"> 1. Customer initiates the booking request. 2. System checks room availability for the desired dates and room type. 3. System reserves the room based on availability. 4. System saves reservation details. 5. System sends confirmation to the user.
Post-condition	A room reservation exiss in the system.

Use Case: Retrieve Customer Details (FR-06)

Primary Actor	Customer
Goal	Retrieve stored customer contact information.
Pre-condition	A customer account exists in the system.
Main Flow	<ol style="list-style-type: none"> Customer opens the system and selects "Retrieve Details." System retrieves the stored contact details of the customer.
Post-condition	The contact details of the customer are displayed.
Post-condition	A new customer account exists in the system.

Use Case: Edit Reservation (FR-05)

Primary Actor	Customer
Goal	Modify an existing reservation.
Pre-condition	Reservation is confirmed.
Main Flow	<ol style="list-style-type: none"> Customer accesses the system. System displays the confirmed reservation. Customer edits reservation details and confirms changes. System updates the reservation.
Post-condition	The reservation is updated.
Post-condition	The reservation is updated.

Use Case: Cancel Reservation (FR-05)

Primary Actor	Reservation
Goal	Cancel an existing reservation.
Pre-condition	A reservation exists in the system.
Main Flow	<ol style="list-style-type: none"> User initiates a request to cancel a reservation. System cancels the reservation. System notifies the user of the cancellation.
Post-condition	The reservation is cancelled.
Post-condition	The reservation is cancelled.

Use Case: Check-in Guest (FR-07)

Primary Actor	Receptionist
Goal	Complete the guest check-in process.
Pre-condition	A reservation is confirmed.
Main Flow	<ol style="list-style-type: none"> 1. Receptionist retrieves reservation details. 2. Receptionist records the guest as checked-in. 3. Receptionist provides the guest with room access.
Post-condition	The guest is checked into the system.
Post-condition	A guest is checked into the system.

Use Case: Check-out Guest (FR-08)

Primary Actor	Reservation
Goal	Complete the guest check-out process.
Pre-condition	Customer is checked in and has an active reservation
Main Flow	<ol style="list-style-type: none"> 1. Receptionist selects "Check-out Guest." 2. System processes the check-out. 3. System updates the reservation status.
Post-condition	Reservation status is updated in the system to checked-out.
Post-condition	Reservation status is updated in the system to checked-out.

Use Case: Process Payment (FR-09)

Primary Actor	Payment
Goal	Complete the payment process.
Pre-condition	Pending payment exists in the system.
Main Flow	<ol style="list-style-type: none"> 1. System retrieves payment details. 2. System processes the payment. 3. System marks the payment as completed. 5. System sends completed.
Post-condition	The payment is completed.
Post-condition	The payment is completed.

Use Case: Issue Invoice (FR-10)

Primary Actor	Customer
Goal	Receive an invoice for a stay
Pre-condition	The customer has checked out of the stay.
Main Flow	<ol style="list-style-type: none"> 1. System processes the check-out of the customer. 2. System generates an invoice. 3. System issues the invoice to the customer. 4. System closes the payment process.
Post-condition	The customer has received an invoice for the stay.
Post-condition	A new customer account exists in the system.

Use Case: Send Notifications (FR-11)

Primary Actor	System
Goal	Send notifications to customers regarding reservation updates.
Pre-condition	A valid reservation exists in the system.
Main Flow	<ol style="list-style-type: none"> 1. System detects a reservation-related event (created, updated, cancelled, or payment completed). 2. System retrieves the customer's contact information. 3. System generates the appropriate notification message.
Post-condition	Notification is delivered and stored in the system log.
Post-condition	Notification is delivered

Use Case: View Monthly Reports (FR-11)

Primary Actor	Hotel Manager
Goal	Access monthly reports.
Pre-condition	System must contain monthly reports.
Main Flow	<ol style="list-style-type: none"> 1. Hotel Manager logs into the system. 2. Hotel Manager selects the "View Monthly Reports" option. 3. System retrieves the monthly reports. 4. System displays reports to the Hotel Manager.
Post-condition	Amew manager has viewed the mohthy reports.

This is the last use case:

Use Case: Print Reservation (FR-06)	
Primary Actor	Customer
Goal	Generate a hard copy of reservation details.
Pre-condition	An existing reservation is present.
Main Flow	<ol style="list-style-type: none">1. Customer logs into the booking system.2. Customer navigates to their reservation.3. System retrieves the reservation details.4. Customer selects "Print Reservation."
Post-condition	A hard copy of the reservation details is generated.
Post-condition	A hard copy of the reservation details



Acceptance Criteria — scenario → measurable expected result.

Registration:

Valid inputs submitted → Account stored in User DB; success message displayed.

Existing email/phone → “Account already exists” error displayed.

• Login:

Valid credentials → User dashboard loads successfully.

Invalid credentials → Specific “Invalid email or password” error message displayed.

• Booking:

Available room selected and confirmed → Reservation created with status Pending Payment.

No available rooms → “No rooms available” message displayed.

• Payment:

Valid Credit Card / approved Installment → Reservation status updates to Confirmed.

Payment declined → “Payment Failed” message shown; status remains Pending Payment.

• Modification (Edit/Extend):

Request to modify reservation → System checks availability.

If available → New dates updated successfully.

If unavailable → “Selected dates not available” message displayed.

• Check-in:

Customer identity verified → Reservation status updates to Checked-in; room assigned.

• Check-out:

Check-out confirmed → Status updates to Checked-out; invoice generated and printed.

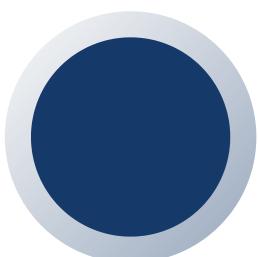
• Reports:

Manager requests monthly report → System retrieves data and generates report showing totals and statistics.

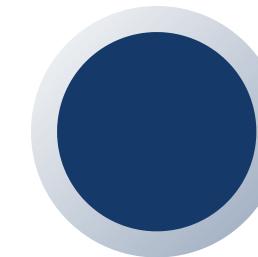
Traceability Matrix

FR-ID	Use Case	Sequence Step(s)	Class	Activity Node	State Transition
FR-01	N/A	Submit login request → Validate credentials	Receptionist (login()), Staff	Enter Login Data → Validate Login	N/A
FR-02	Create Account	Fill registration form → Save new user account	Customer(createAccount())	Display Registration Form → Create Account	N/A
FR-03	Specify Room & Dates	Check availability → Return status	Room (isAvailable()), Room Type	Check Availability → Room Availability = True	N/A
FR-04	Make Reservation	Send booking request → Save reservation	Customer, Reservation	Book Room → Create Reservation	Created → Pending Payment
FR-05	Edit/Cancel/Extend Reservation	N/A	Reservation (editReservation(), cancelReservation())	Edit → Modify Details / Cancel / Extend → Update Dates	Pending Payment → Cancelled Confirmed → Confirmed
FR-06	Retrieve Customer Details	Retrieve customer data → Return profile	Customer (name, email, contactNumber)	Verify Customer Identity	N/A
FR-07	Check-in Guest	N/A	Reservation (checkInDate), Room	Set Status = Checked_in → Assign Room	Confirmed → Checked-in
FR-08	Check-out Guest	Send check-out request → Confirm check-out	Receptionist, Reservation	Initiate Check_out → Confirm Check_out	Checked-in → Checked-out
FR-09	Process Payment	Process payment → Update payment success	Payment, CreditCard, Installment methods	Enter Payment Info → Payment Success	Pending Payment → Confirmed
FR-10	Issue Invoice	Print invoice & thank-you message	Invoice (generateInvoice()), Receptionist	Generate Invoice Bill	N/A
FR-11	Send Notifications	N/A	Reservation (sendNotification())	Send Confirmation Via Email	N/A
FR-12	View Reports	N/A	HotelManager (viewMonthlyReports())	Request Monthly Report → Generate Monthly Report	N/A
FR-13	Payment Encryption	N/A	Payment	Enter Payment Information	N/A

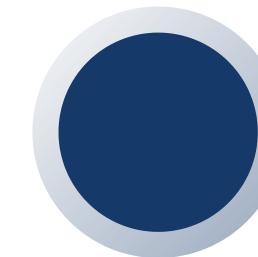
4. Analysis Models (UML)



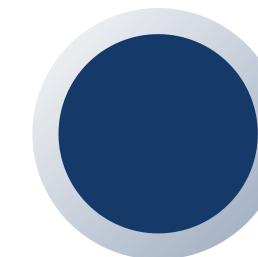
4.1 Use Case Diagram



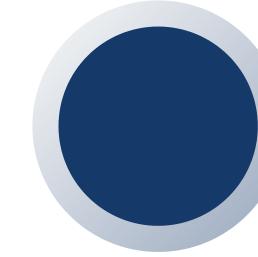
4.2 Class Diagram :



**4.3 Sequence Diagrams –
Main & Exception**

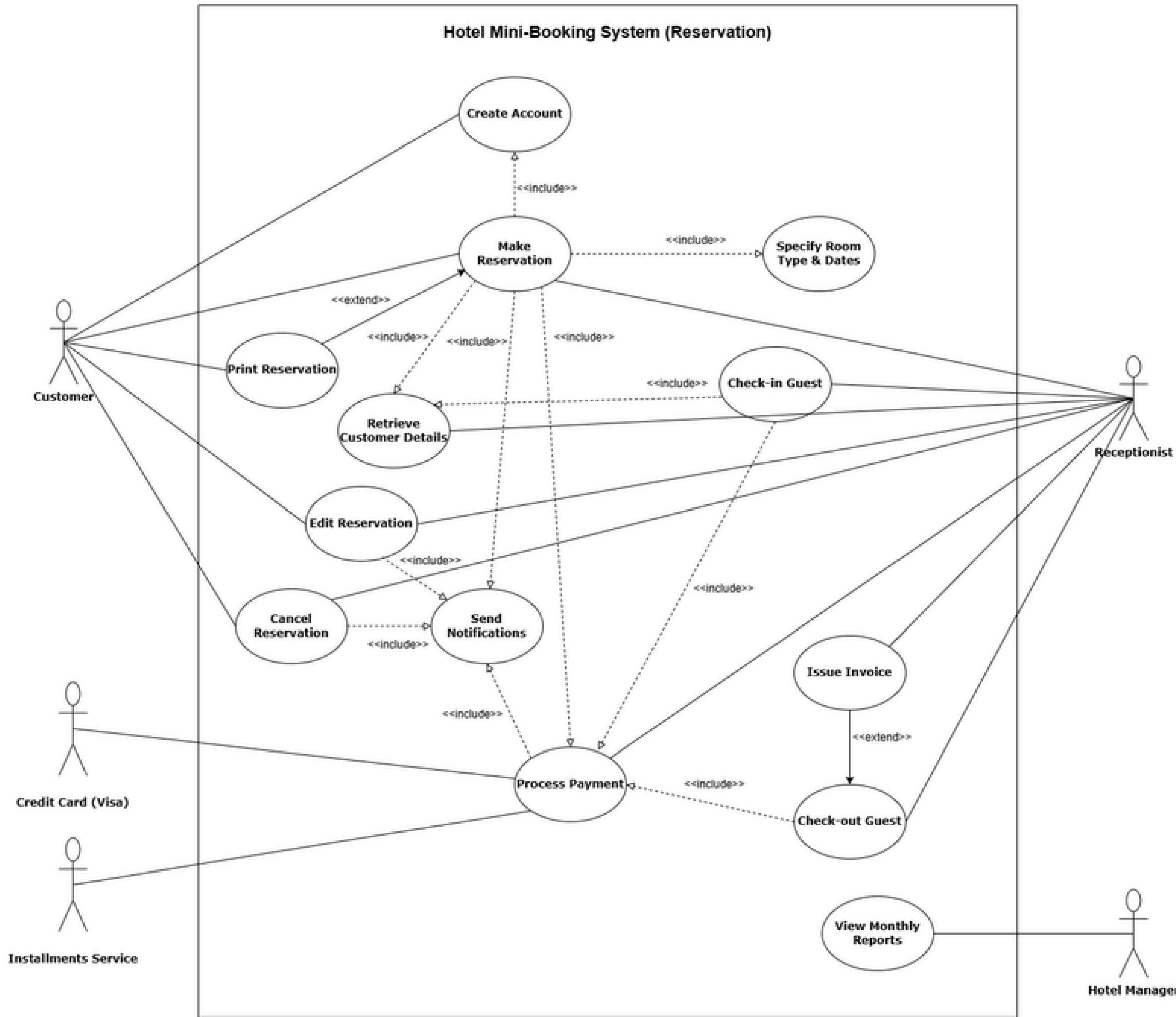


4.4 Activity Diagram

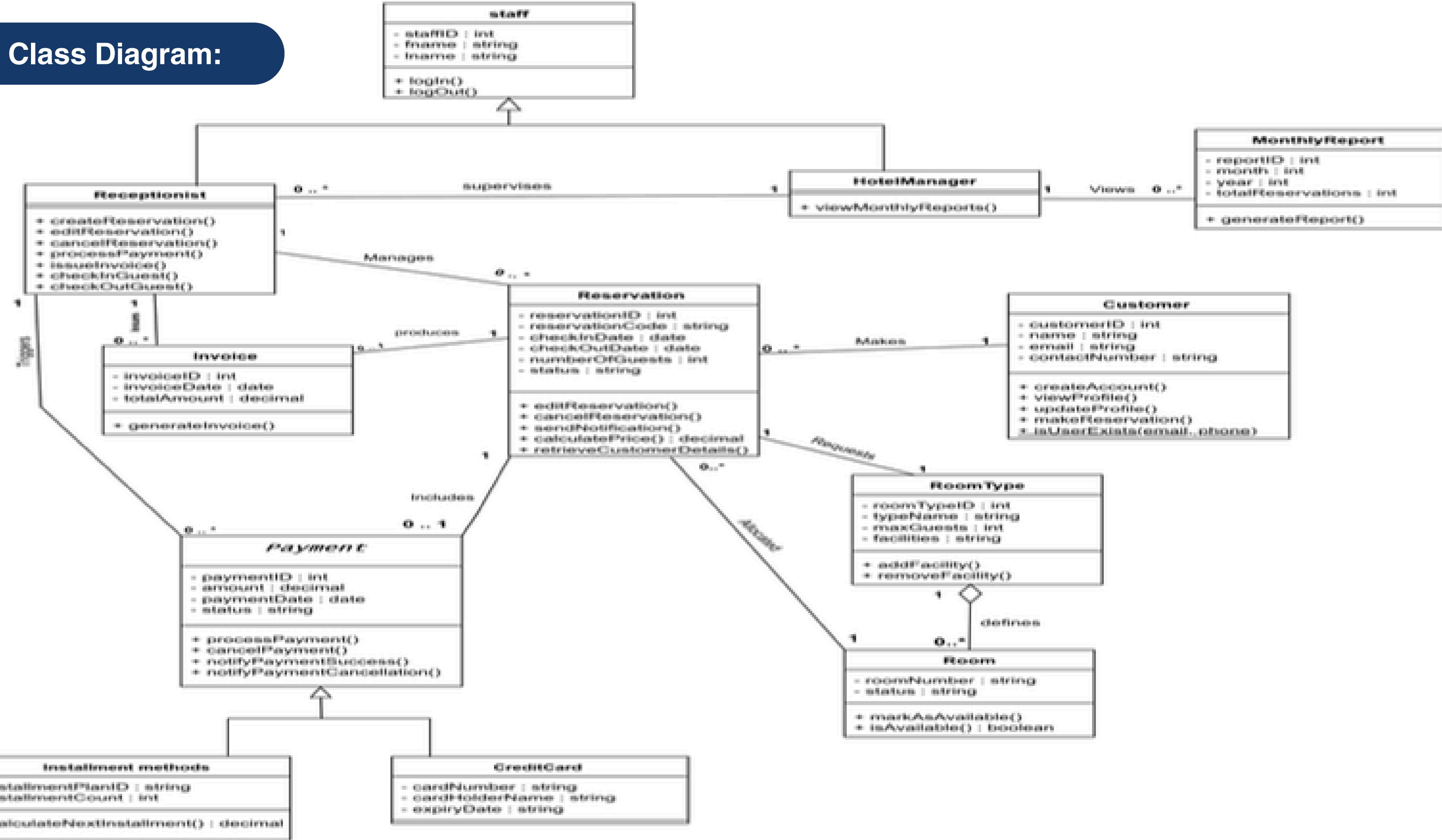


4.5 State Diagram

Use Case Diagram

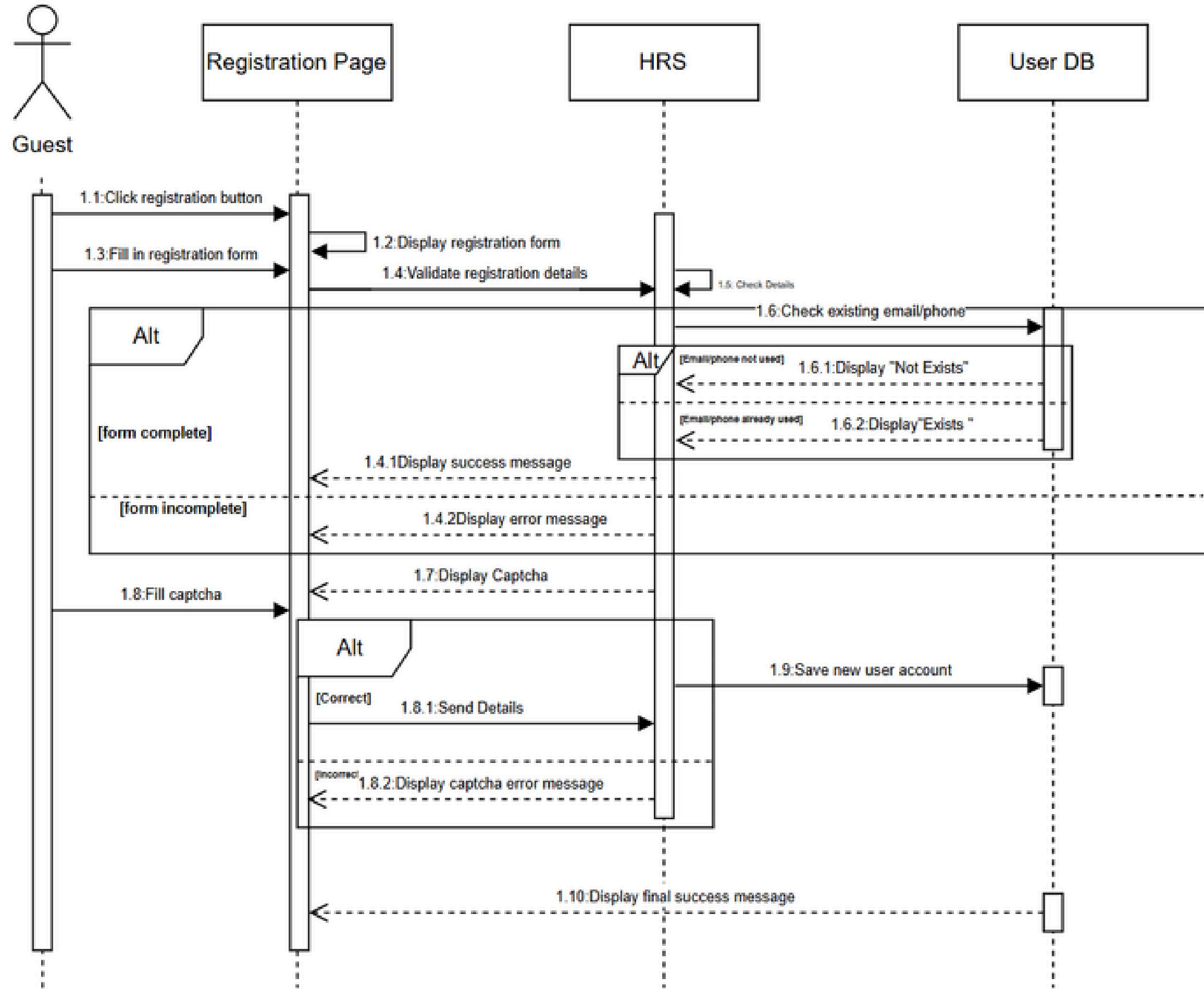


Class Diagram:



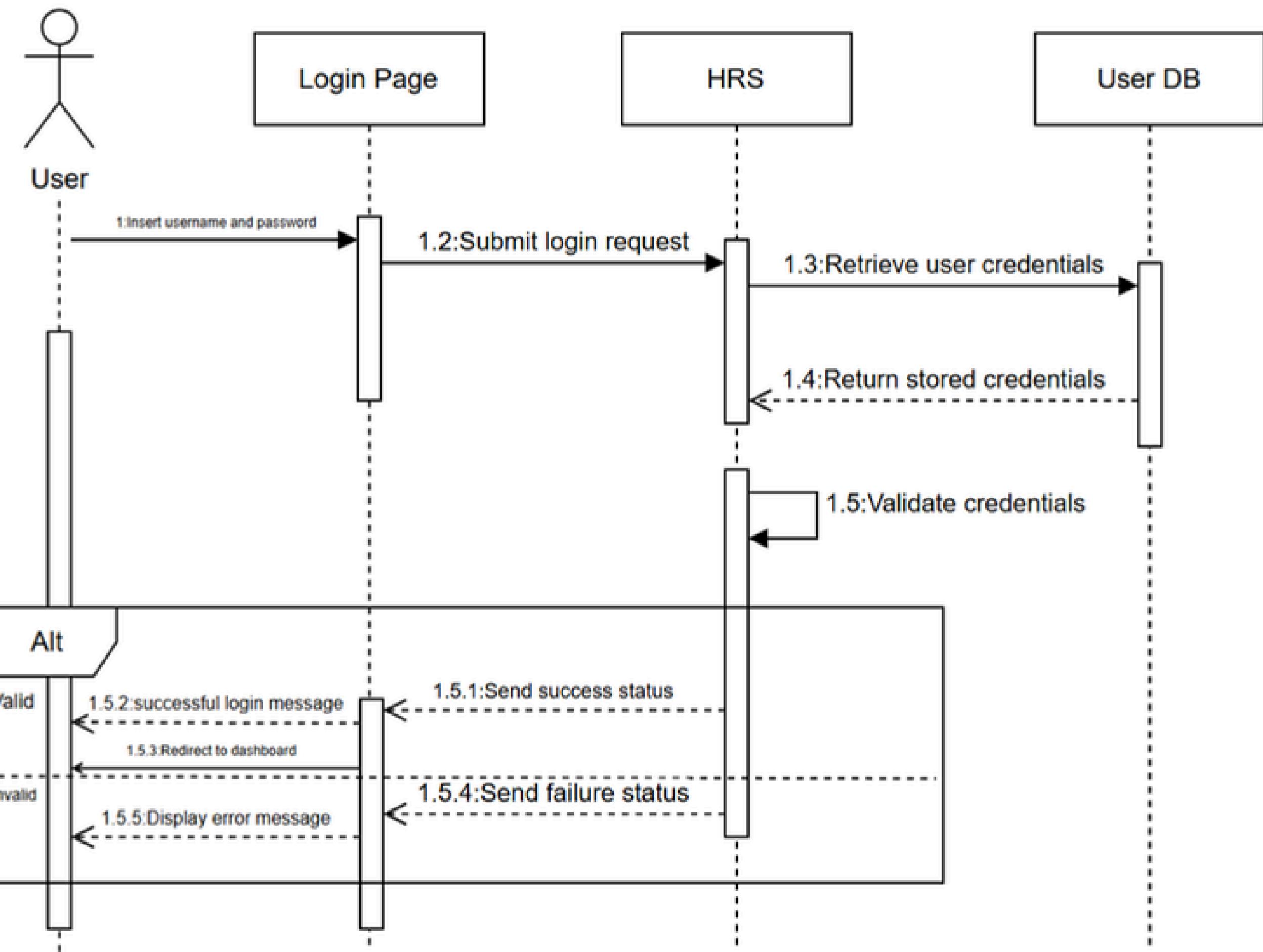
Sequence Diagrams

**Create New Account
(Customer Registration)**



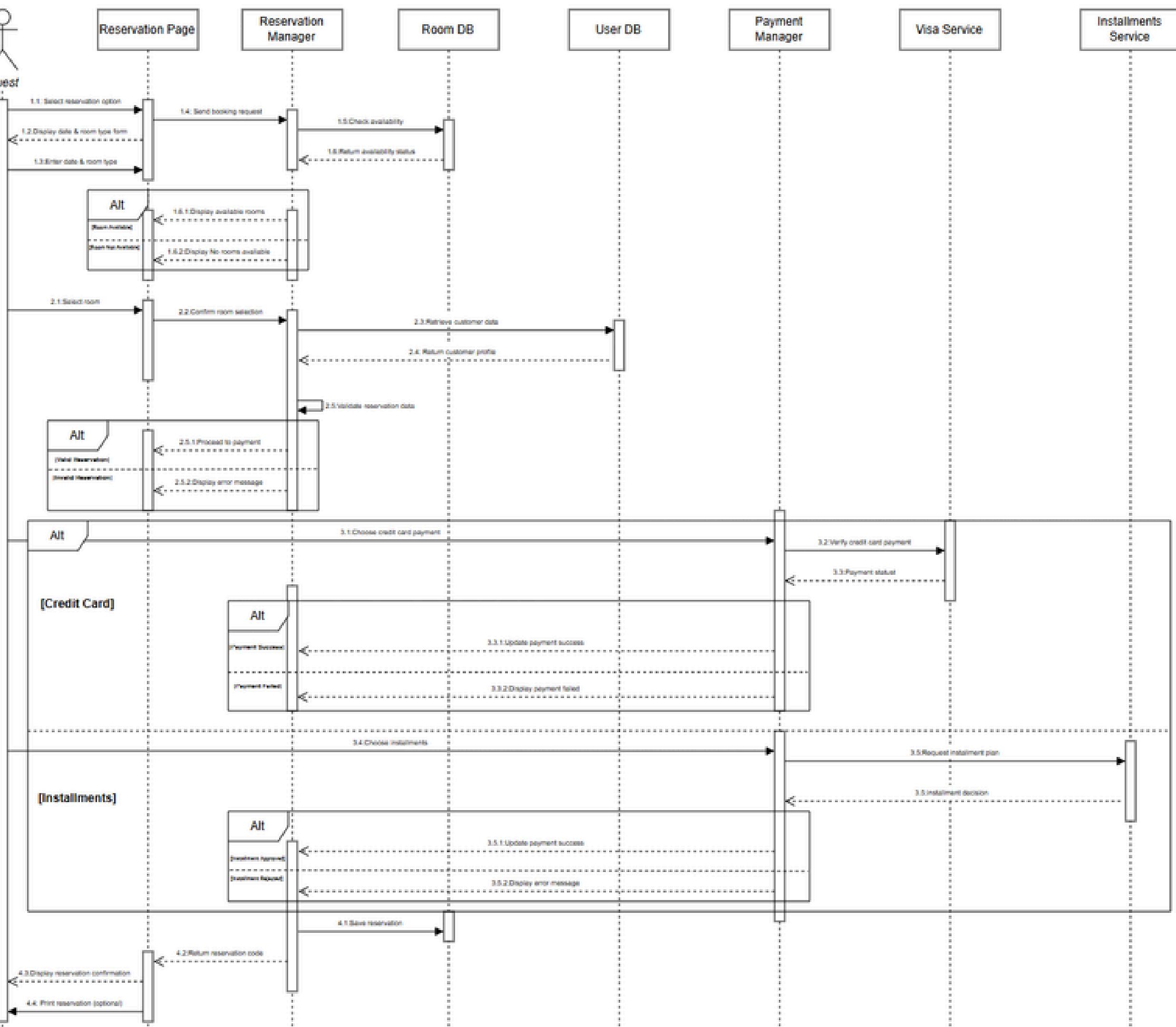
Sequence Diagrams

Customer Login



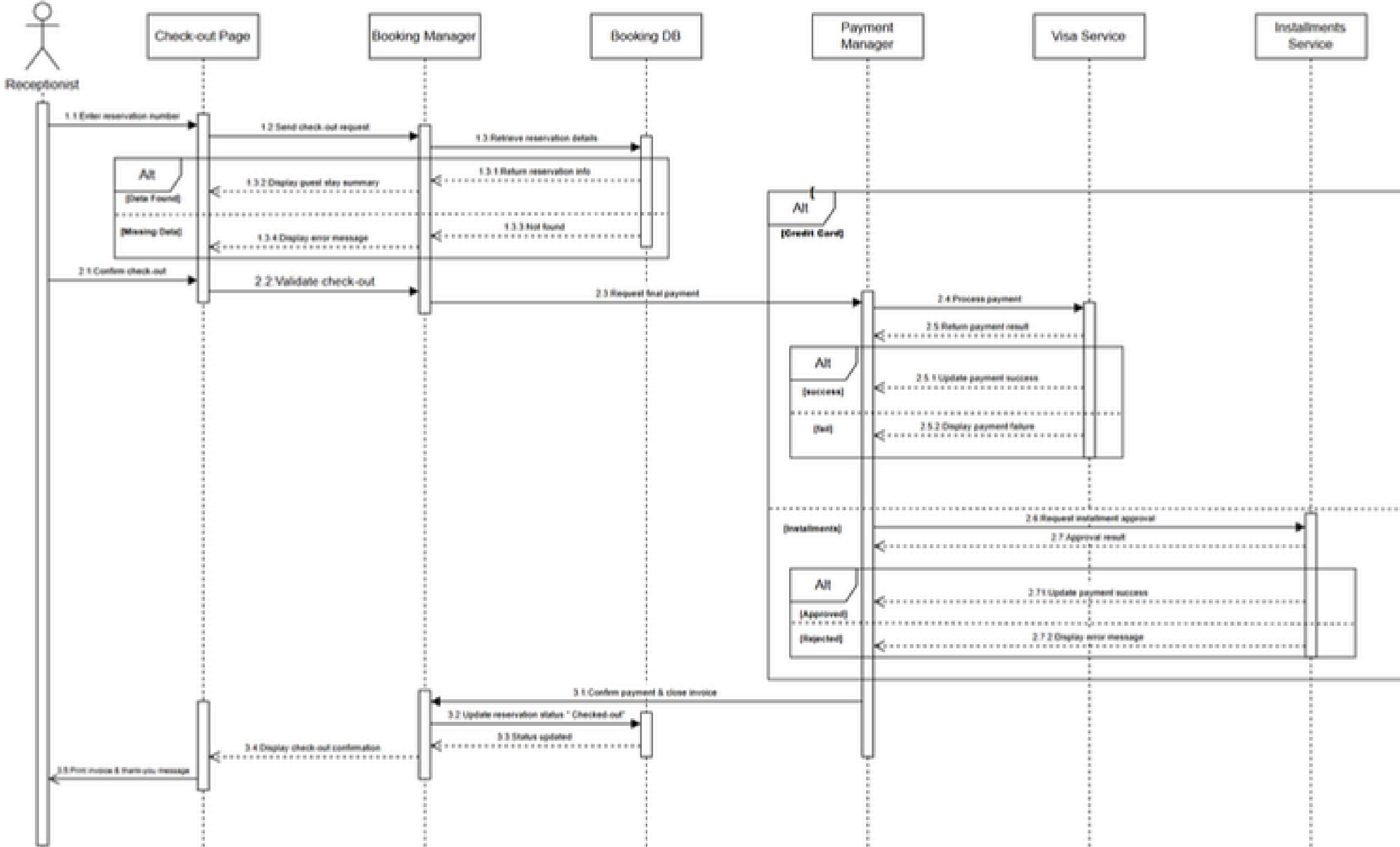
Sequence Diagrams

**Make / Manage
Reservation and
Process Payment**



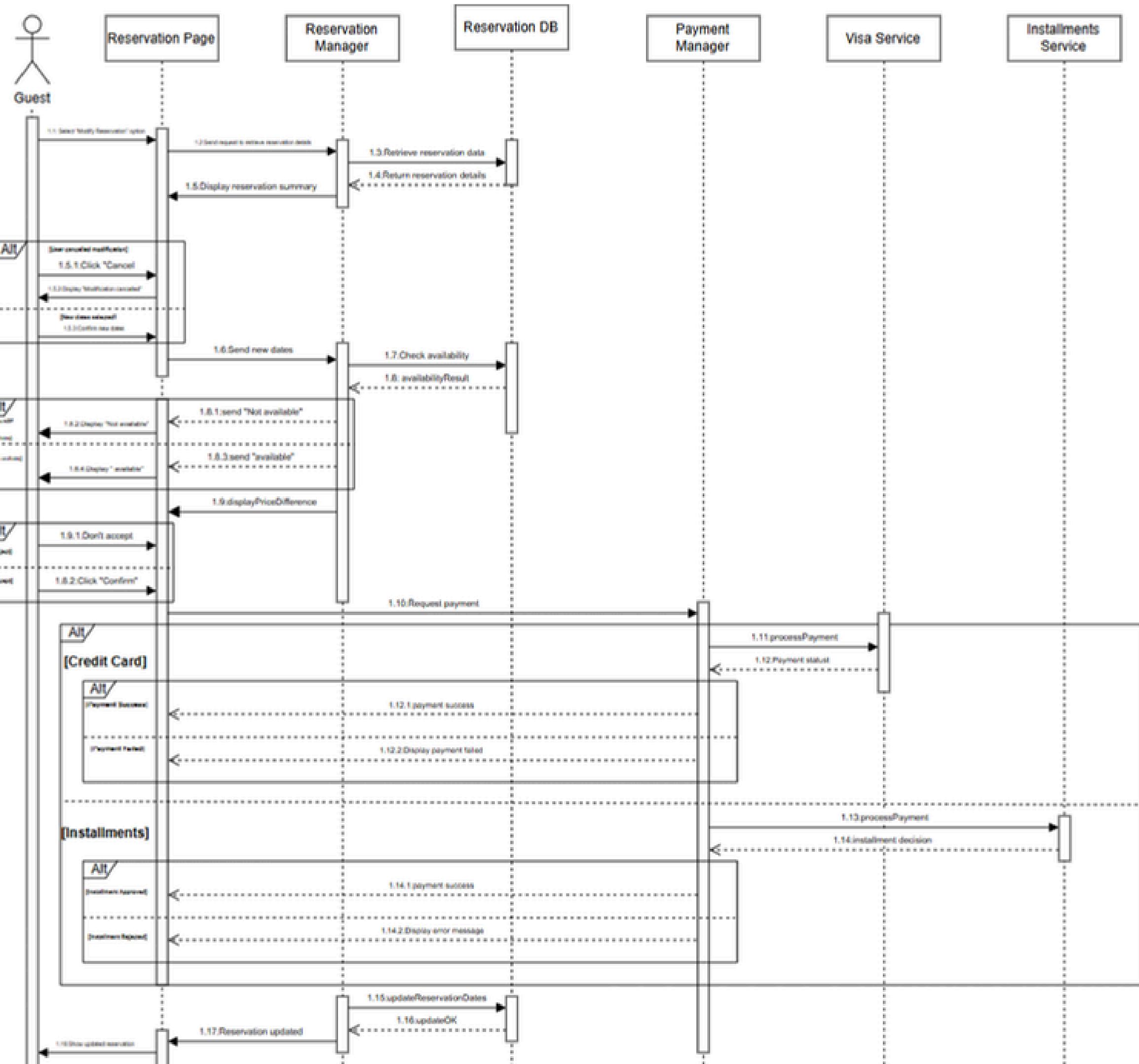
Sequence Diagrams

Guest Check-out



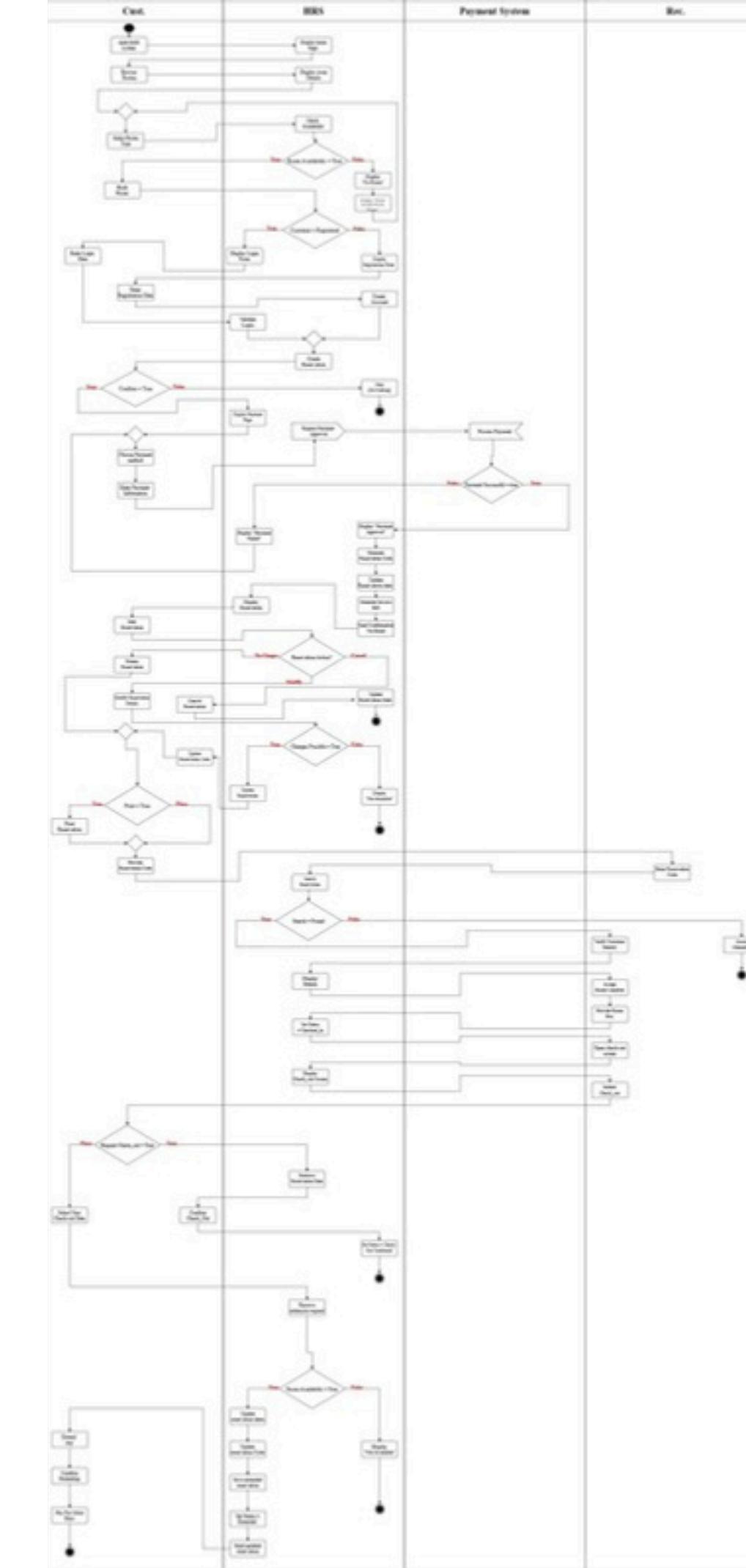
Sequence Diagrams

**Extend Reservation
(Modify Reservation
Dates)**



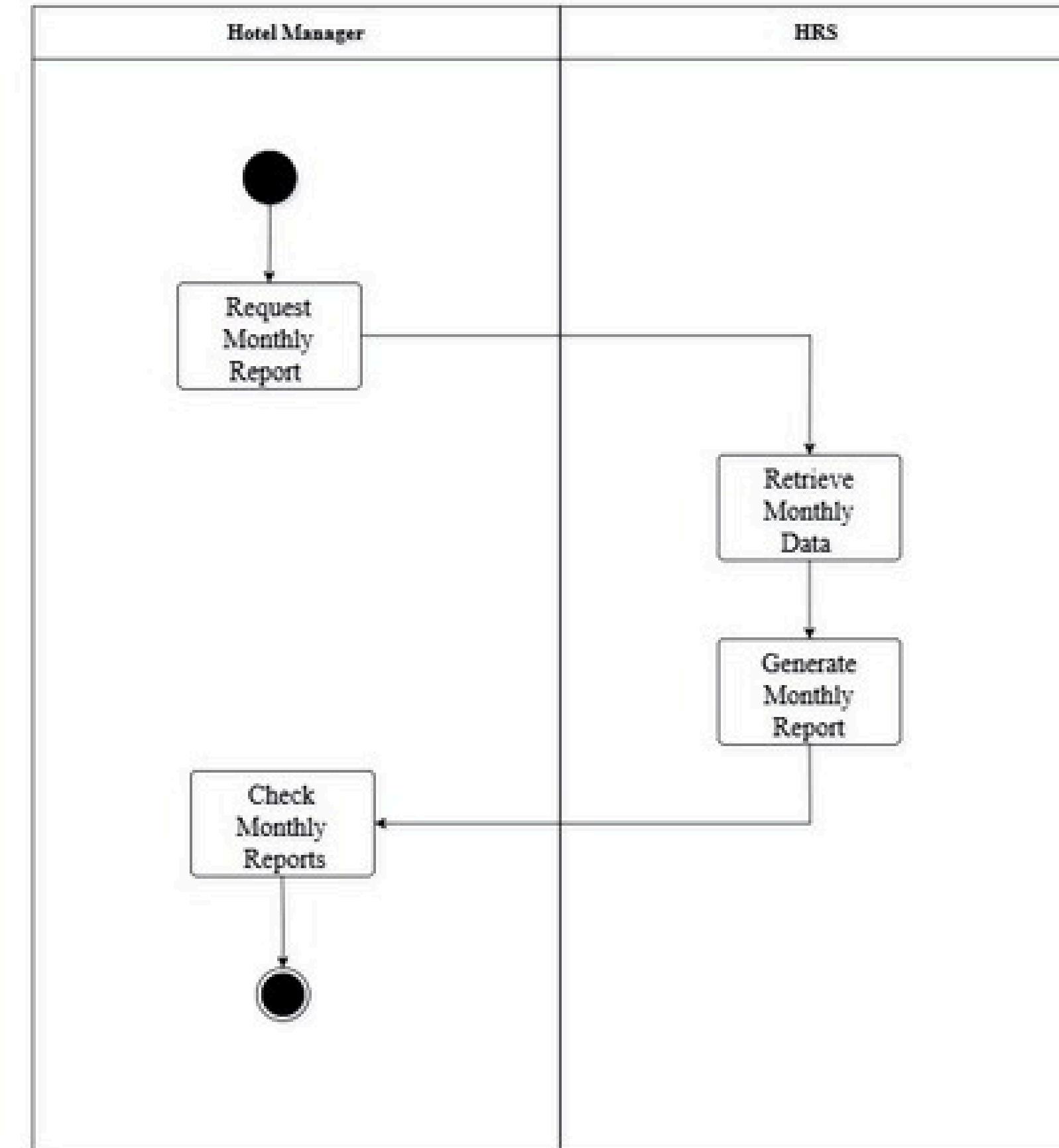
Activity Diagrams

**HRS Activity
Diagram**

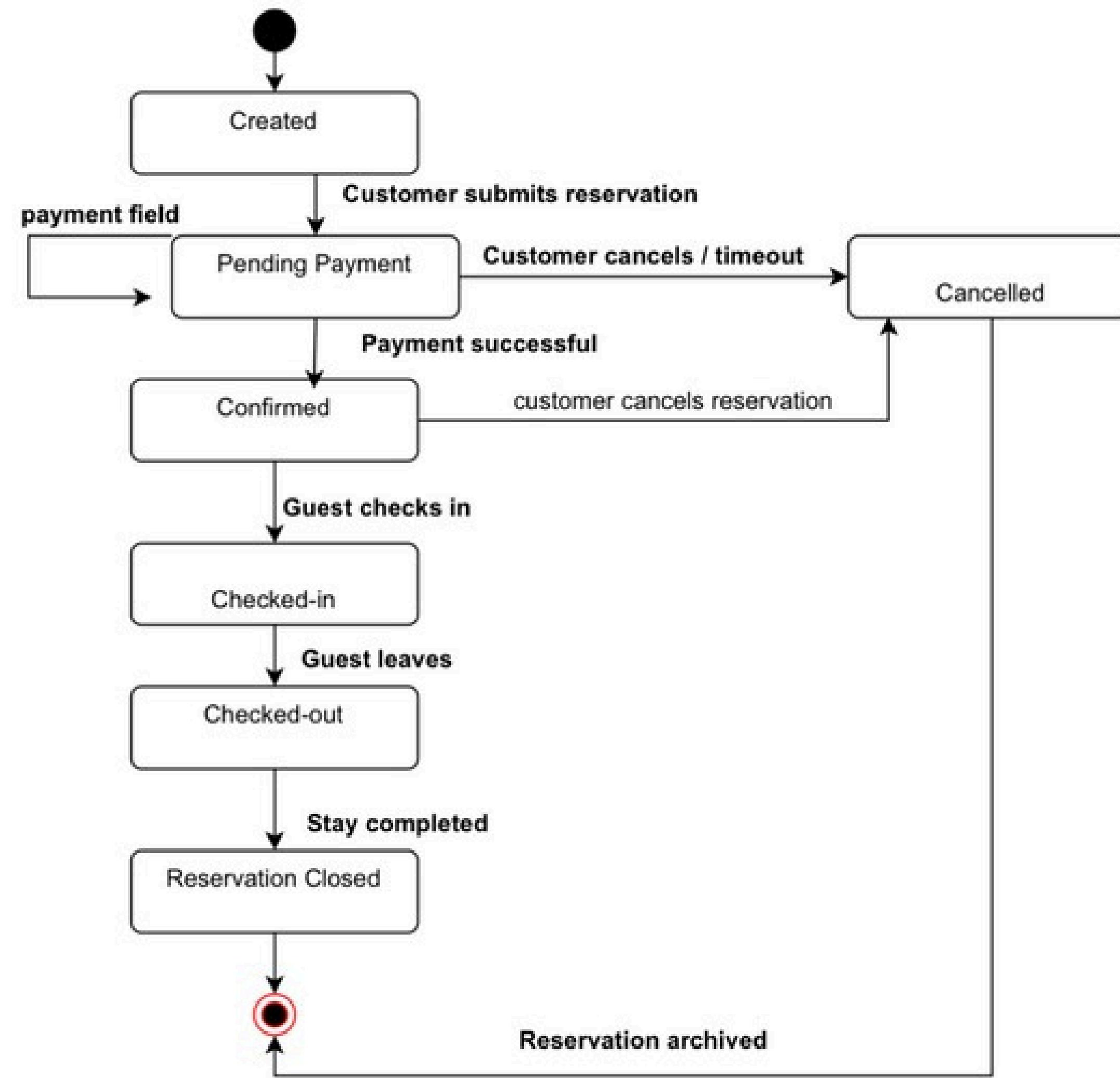


Activity Diagrams

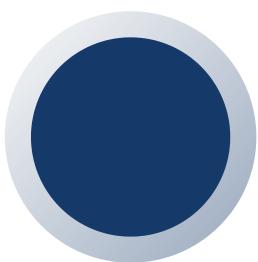
***Reports Activity
Diagram***



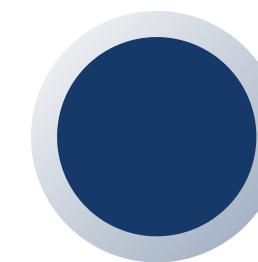
State Diagram



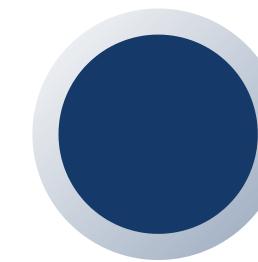
5. Change Management Process



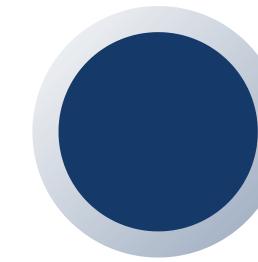
5.1 Who Can Request Changes



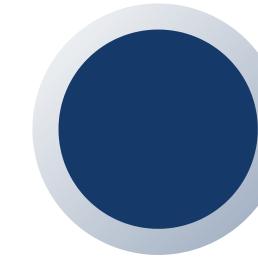
5.2 Logging Change Requests



5.3 Review and Approval



5.4 Version Control



5.5 Summary

Who Can Request Changes

This section describes how changes to the Hotel Mini-Reservation System are requested, reviewed, approved, and applied to all project documents.

The following stakeholders may submit a change request:

- Project team members
- Instructor / Product Owner
- System users (Customer, Receptionist, Hotel Manager)
- QA/Testers during system testing

Logging Change Requests

All change requests must be recorded in the Change Log and include:

- CR-ID
- Title and description of the change
- Reason for the change
- Priority (Low / Medium / High / Critical)
- Requested by
- Impacted items:
 - Functional Requirements (FR-01 → FR-13)
 - If more FRs are added later (e.g., FR-14), update the range accordingly.
 - Non-Functional Requirements (NFR-XX)
 - Use Cases
 - Class Diagram
 - Sequence Diagrams (all 4)
 - Activity Diagram
 - State Diagram

Review and Approval

- The Team Leader (Maznah Hadi) performs an initial review.
- The team conducts an impact analysis on all affected requirements and UML models.
- The change is either approved, rejected, or returned for clarification.
- If approved, all affected diagrams and requirements are updated to maintain consistency.

Version Control

- The SRS uses the following version format:
- Major.Minor.Revision
- Examples:
- 1.0.0 – Initial version
- 1.1.0 – Updated UML or FR
- 1.1.1 – Minor corrections
- A version history table is updated after each approved change.

Summary

All approved changes must be reflected in:

- Functional Requirements
- Non-Functional Requirements
- Use Cases
- All UML diagrams in Section 4:
 - Use Case Diagram
 - Class Diagram
 - Sequence Diagrams
 - Activity Diagram
 - State Diagram
- Test cases and acceptance criteria must also be updated to match the new or changed requirements.

This ensures the SRS remains accurate, consistent, and up to date.

Thank You!