WACHEMO UNIVERSITY

COLLAGE OF ENGINEERING AND TECHNOLOGY

DEPRTMENT OF SOFTWARE ENGINEERING

PROJECT TITLE: - HOTEL ROOM RESERVATION SYSTEM

YEAR: - SECOND YEAR FIRST SEMESTER

COURSE: FUNDAMENTALS OF SOFTWARE ENGINEERING AND DATA BAS

GROUP 6

NAME	ID
1.BEIMNET BELAY ORSIDO	WCU170232
2.SIRAWDINK YOHANNES LAMBEBO	WCUD6703
3. ABENEZER TAMIRAT ABIYE	WCU179618
4. FITSUM GONDER SOKIYAB	WCU174931
5.MERIHUN MESHESHA MARA	WCU17D3530

SUBMITTED TO: MS SENDU GM

SUBMISSION DATE: 17 /04/2018 E,C

ACKNOWLEDGMENTS

Special thanks for our adviser MS SINDU GM and MS FOZIA ABAKO For continuous support in every phase of the project and who continuously provides us his valuable advice to prepare this project document. Also thanks to the shewaber hotel worker who gives for us a correct requirement for the system developed by our group members Our Appreciation also extends to our instructor 's who helps for us to develop this project Lastly, we would like to acknowledge all our friends for their morals, idea, materials and financial support during the courses of the semester.

TABLE OF CONTENT

LIST OF TABLE

LIST OF FIGURES

ACRONOMY

HMS – Hotel Management System

· SRS – Software Requirement Specification (implied in document)

· UML – Unified Modeling Language

· ERD – Entity Relationship Diagram

· DFD – Data Flow Diagram

· SQL – Structured Query Language

· DBMS – Database Management System

· RDBMS – Relational Database Management System

· API – Application Programming Interface

· ACID – Atomicity, Consistency, Isolation, Durability

· PK – Primary Key

· FK – Foreign Key

· NF – Normal Form (e.g., 1NF, 2NF, 3NF)

· UI – User Interface

· UX – User Experience

· QA – Quality Assurance

· UAT – User Acceptance Testing

· POS – Point of Sale

· RBAC – Role-Based Access Control

· LAN – Local Area Network

· UPS – Uninterruptible Power Supply

· NIC – Network Interface Card

EXCUTIVE SUMMARY

The current system at Shewa Ber Hotel relies on manual and partially computerized processes for managing customer records, room bookings, and payments. This approach results in inefficiency, data inconsistency, security risks, and slow service delivery.

The proposed Shewa Ber Hotel Management System introduces a fully computerized solution that automates hotel operations through a centralized database and a three-tier architecture. The system improves accuracy, security, efficiency, and reporting while enforcing business

CHAPTER ONE : INTRODUCTION

A Hotel Management System is usually referred to as a Property Management System, is a complete, integrated software platform designed to efficiently organize and automate all aspects of daily hotel business operations. It is also useful for rapid,efficient and reliable manner for the hotel business and it reduces the need for fragmented systems and manual processes.

The main objective of this project is to provide a solution for hotels to manage most of their work using computerized processes. This software will help admin to handle customer information, room allocation detail, payment details and the room have different categories like single bed, double bed etc. so their record will be maintained accordingly.

1.1 ORGANIZATION BACKGROUND

ShewaBer Hotel is one of the well known customer services establishments located in the rapidly growing town of Hossana, in the Hadiya Zone of Southern Ethiopia and Formed in 1986 E.c one of the oldest hotel in hossana, and it formed by Geredo Workicho and his family currently his youngest brother ashagre werkicho is take his place and manage the hotel .the hotel was established to meet the increasing demand for quality accommodation and customer oriented hospitality services. Over the years ShewaBer Hotel has gained recognition for its commitment to providing clean, comfortable, and affordable services to both local and visiting (e.g. : zemen bus passenger) guests by using manual system. Initially starting with a limited number of rooms and basic lodging services. Shewa Ber Hotel gradually expanded its facilities in response to the growing number of travelers, government workers, business visitors, and families coming to Hossana for various purposes.

Recognizing the dynamic nature of the hospitality industry, the management has continuously invested in improvements such as upgrading guest rooms and management.as well that rooms are properly managed and that guest receive timely and introducing better sanitation and safety measures. These efforts have helped the hotel adapt to customer expectations and stay relevant in a competitive market. While the whole hotel workers and members are very committed and putting in a great effort , the lack an of automated system affect their productivity and that is the customer experience. This software requirement specification document is identifying the requirement for a modern integrated system that can address such operational challenges better while ensuring high efficiency and scalability, together with increased customer satisfaction.

1.2 STATEMENT OF THE PROBLEM

Shewa Ber Hotel, a recognized guest service establishment in Hossana, faces several operational and managerial challenges that limit its ability to deliver consistent, high-quality services. due to the hotel using manual system for many purposes.

For example: using paper, spreadsheets for storing data and guest information.

So there are many challenges These challenges include gaps in staff skills and training, limited modern facilities, inconsistent customer service standards, and underdeveloped marketing strategies. With the rapid growth of hotels and lodges in Hossana, these issues pose a risk to the hotel's competitiveness, customer satisfaction, and long-term sustainability. Conducting a detailed study of the hotel's organizational structure, operational processes, and service delivery is therefore essential to identify weaknesses and propose practical solutions. This project is justified as it provides insights to improve service quality, enhance customer experience, strengthen competitiveness, guide managerial decisions, and contribute to both the academic understanding of hospitality management and the local economic development of Hosanna.

1.3 OBJECTIVE OF THE PROJECT

1.3.1 GENERAL OBJECTIVE

Develop hotel management systems of shewaber hotel to improve service quality and productivity. and also to reduce paperwork and manual system in order to achieve maximum guest satisfaction and make profit by providing good service managing all resource productively .

1.3.2. SPECIFIC OBJECTIVE

The specific objective of the project are mention as follow :

- Reviewing how the current system work and operate.
- Investigate how the existing system is operating.
- Design a new system that can overcome the problem of the current system .
- Identify functional and non functional requirement for the new system .
- Develop and implement the new system .

1.4 METHODOLOGIES

1.4.1 REQUIREMENT ELICITATION METHODOLOGY

- The following are some of the methodologies that were use gather the requirement for defining the shewaber hotel management system.

1.interview : through structured interview with the manager and the owner of hotel about the work flow problem faced and functionality desired.
2.Questionnaire **:** a structured research instrument consisting of a set of standardized questions used to systematically gather data, feedback, or requirements from stakeholders, users, or developers.

- ✓ **mainly we used questionnaire**

REQUAREMENT GATHERING TEMPLATE

- **Project and Stakeholder information**

Background of a Hotel	Shewaber hotel Formed in 1986 E.c one of the oldest hotel in hossana, and it formed by Geredo Work icho and his family currently his youngest brother ashagre werkicho is current owner
Interviewee Name	Ashagre Workicho owner and Getachew Mekonin (manager)
Hotel name	SHEWABER HOTEL
Address	Shewaber located along the main street of city of hosaena ,normally called mariam sefer ….
Phone no	0916757562
Date	13/03/2018
Interviewer	Beimnet belay and sirawdink
what are the primary goal to develop this system?	the primary goal is to automate current manual system in order to improve guest satisfaction ,increase profitability ,and to improve efficiency and modern hospitality for a gust.

Are you currently using any software? or Is the system manual ?	No, all thing by manual system
What do you like about your system ?	The current system is easy to understand , direct management ,accessibility , adaptability ,it's security since data is not stored on network ,it is not attacked by hackers…..
What are the biggest problem with your current system ?	Using paper , spreadsheets for storing data and guest information ,extreme inefficiency. Manual system force the hotel react slowly
What permission should each role have ?	They have all different role in the system
what is the process for assigning rooms?	First they make sure the room is available, clean ,and ready for a new guest if the room occupied or full so they can't accept new guest . Then also the room need to be cleaned before it can be sold again or reserve again
What information need to be store for each room type ?	Room no, bed type (is the room have two bed ,or it have only single bed?), view and price.
What are the payment methods the system need to support ?	Customer can pay with mobile banking, internet banking and also use other different payment options
What type of report are needed ?	Room occupied, available, revenue ,guest statistics(how much day guest can stay)..

TABLE 1.1REQUAREMENT GATHERING TEMPLATE

3.observation: observed the operation of shewaber hotel in relation to inefficiencies in their current manual process of handling order and manual system .

4.Document Analysis -Analyzed available collect records to understand the flow of information and how things are presently done.

5.**Focuse group** -discussion with the staff to identify pain points and gather insights into features that could improve their efficiency and reduce work load.

1.4.2 REQUIREMENTS ANALYSIS AND MODELING

-In this stage once, collect, organized and cleanly understanding of functional and non functional requirement of the system.

Requirement Analysis

1. functional

- Automate the billing system for speed and efficiency in customer transactions in different way.
- Provide complete reports on sales, inventory and staff performance.
- Manage the room available ,clean and ready for new gust.
- Real time inventory tracking need to be followed in order to avoid wastage.

2. non functional

- -Usability -the system should be easy to use by personal with out no experience .
- -Reliability-the system to perform its required functions correctly, consistently and with out failure for specific period of time.
- -security -with the ability to access secret information related to the financial documentation .

- **Modeling**

Models to visually explain and performance analysis on the system requirement.

1. Use case diagram : model show the interaction between user(actor) and the system.

2. Activities diagram : model shows the work flow or business processes .

3.Class diagram (structural model) : show the static structure of the system.

4.Sequence diagram : shows how object interact in a time sequence for a specific use case.

5.Entity relationship diagram: it show that data entities and their relationship.

6.state chart diagram :it show its states and the transitions between them due to events.

7.component diagram : it show break down large object oriented system in to smaller components.

8.Deployment diagram :it show system hardware , software and network connections for distributed computing.

1.4.3. SYSTEM IMPLEMENTATION METHOD

The system will be developed using an iterative development methodology, in which design, implementation, and refinement of functionalities are conducted through successive development cycles. Each cycle comprises planning, coding, testing, and evaluation, enabling early detection of issues, incorporation of feedback, and continuous enhancement of system capabilities. This method ensures that the system remains flexible and responsive to evolving requirements and unforeseen challenges during development.

1.5 SCOPE AND LIMITATION OF THE PROJECT

1.5.1 SCOPE

The scope of shewaber hotel management system is to improve simplify and manage hotel operations efficiently, Accurately and keep customer happy. Generally the scope of shewaber hotel are

A. Customer information management
B. Manage Payment system
C. Room management
D. Reporting and Analytic

1.5.2 LIMITATION

Although the shewaber hotel management system might have been devised to avail wholesome in the management of hotel operation their are a number of limitations it may face in implementation and performance. Some of these are:-

- With out internet the system may not function properly.
- If staff are not fully trained the y may be face difficulties to understand the system.

1.6. SIGNIFICANCE AND BENEFICIARIES OF THE PROJECT

Hotel Ownership & Management:

· Increased Profitability : the system is highly effective for reduced labor cost it means fewer staff are needed for daily task .

. Reduce Errors : the current manual system can easily affected by miss calculated bills that error drive the owner in to losses so the system can prevent this issue.

· Management : the system can automatically adjust room rate based on costumer need and also it easy to manage different room type at once . Hotel Staff :

・Improved hospitality: Automates repetitive manual tasks, freeing up staff to provide better guest service.

・Reduced Errors: Digital records eliminate illegible handwriting and calculation mistakes.

. decrease repetitive tasks : the system automate duplicate manual tasks like updating the room statues(i.e. :clean ,occupied ,available…) ,calculating bills and generate report .

・increase customer Satisfaction: Faster check ins and outs, customer can book online it's easy , and personalized service based on stored preferences.

1.7. FEASIBILITY ANALYSIS

1.7.1 OPERATIONAL FEASIBILITY:

it address major headaches for stuff and owner of the hotel like double booking rooms , messy communication between staff members , and manual room check ins.so the staff members will relief from those problems .

The operational benefits are massive and necessary for archive the goal came by developing this soft ware. The key to success is not the software itself but a solid change management plan that includes thorough training and clear communication about how this will make everyone's job easier in the long run.

1.7.2TECHNICAL FEASIBILITY

it give the hotel cloud based solution it mean we can access it through web browsers .also the system will have integrated capabilities it means the system are built to talk to other soft ware via API(APPLICATION PROGRAM SOFT WARE),making it feasible to connect to your accounting software ,payment gateway and channel manager .

The technology is standard and readily available. The feasibility hinges on choosing the right vendor for your size and needs and ensuring you have a stable internet connection.

1.7.3 ECONOMIC FEASIBILITY

the hotel management software process can't work with out any investment like implementation or set up cost ,data transformation or data migration also need cost or investment ,training, possibly new hardware (example : tablet ,computer, printer).” Will this thing pay it self ?”,not really…

But if the system work properly it have significant cost saving method like:

- Reduce labor cost :the system potentially decreases the need for more hires employees as you grow.

- Reduce error: reduce manual booking and billing mistakes saves money.
- Lower third party commission: by your direct website booking easier. And so much more.

1.7.4 Schedule Feasibility

Scheduling feasibility determines whether the proposed system will be completed on the given time or not .whether the scarcity of time given for the project by the internal motivation and potential of the team member of the project , we surly the project will be completed on time .

1.7.5 Legal Feasibility

this cover data privacy ,contract ,and compliancy . the system key consideration will be data security & privacy , we are storing sensitive guest data like (ID,NAME, ADDRESS, PHONE NO) we have to keep it safe customer sensitive information other wise there are money rule and regulations to deal with legally acceptance of the system.

- There are no major legal road blocks if you work with a reputable vendor. The key is to **have a lawyer review the vendor contract** and to confirm the vendor's compliance certifications for data security and privacy before signing anything.

1.8 RISK ASSESSMENT

1. Potential Risks
    - Requirement Changes: Late changes or additions to system requirements may cause delays.
    - Technical Challenges: Unforeseen difficulties in integrating features or resolving bugs could impact timelines.
    - Resource Availability: Unavailability of key team members or tools may disrupt the development process.
    - Budget Constraints: Exceeding the allocated budget could hold back project progress or result in feature compromises.
    - Deployment Issues: Challenges during deployment, such as compatibility or server configuration problems, may delay the system’s launch.
2. Risk Mitigation Strategies
    - Requirement Changes: Ensure detailed requirements are finalized early and strictly manage scope changes using a change control process.
    - Technical Challenges: Conduct feasibility studies, prototype critical components, and allocate time for unexpected technical issues.

- Resource Availability: Maintain a backup plan for key roles and tools, and ensure all necessary resources are secured in advance.
- Budget Constraints: Monitor expenditures closely and prioritize essential features to stay within the budget.
- Deployment Issues: Perform thorough testing, including deployment rehearsals, to identify and address potential problems early.

3. Risk Monitoring and Control

Regular progress reviews and team meetings will be conducted to identify risks early and implement corrective measures.

- A risk log will be maintained to document potential risks, their impacts, and mitigation actions.

4. Justification

   By identifying potential risks and preparing mitigation strategies, the project can proceed smoothly and meet its objectives within the defined timeline and budget.

1.9 DEVELOPMENT TOOLS

The development tool of the Shewaber Hotel Management System requires a set of software, hardware, and technological tools that support system design, development, testing, deployment, and maintenance. The tools selected ensure reliability, security, and scalability to matching the hotel operational requirements.

1.Software Development Tools

A. Programming Languages

C++ it’s Used for building the core system logic such as:

- Room state management
- Booking and reservation processing
- Payment handling

B. Frameworks

Provides a structured environment for back-end development.

Suitable for secure, scalable hotel management systems.

Ideal for modular and enterprise-level applications.

C. Database Management System

- Guest information
- Booking records

- Room states (clean, dirty, occupied, available)
- Payment history
- Reports and logs
- These DBMS support ACID transactions, preventing double booking or data loss.

D. User Interface Development Tools

To build the web-based interface for: system Admin ,Manager and Online booking for guests

E. system design tools

- ➢ UML(unified modeling language)
- ➢ SQL workbench
- ➢ Enterprise architecture

F. Integration Tools

Used for connecting:

- Online booking portal
- Mobile access
- Room and billing modules
- Payment gateway

2. Deployment Tools

A. Local Server / Hosting Tools

For hosting the system locally in the hotel.

B. Cloud Hosting (Optional)

Google Cloud Supports online booking and remote access.

C. Network Tools

Routers, network cables, switches to support:

Local network for reception and manager

Stable internet for online booking

3.Hardware Tools

A. Computers / Laptops or mobile

B. External Storage

Hard drives / Flash disks for backups

Ensures protection against data loss

4.Security Tools

A. Encryption Tools

Database encryption methods to protect guest details.

B. Access Control

Role-based permission configuration in the system:

C. Backup Tools

- Automated backup scripts
- Cloud backup storage

1.10 WORK BREAKDOWN

1.10.1 PROJECT PLAN ACTIVITIES – Schedule the Shewaber Hotel Management System development follows a structured activity plan and designed to ensure smooth progress, quality output, and timely completion.

Activity	Duration	Start date	End date	Responsible person
Requirement gathering & validation	1 weeks	06/03/2018	13/03/2018	BEIMNET AND SIRAWDINK
System design(data base ,work flow model)	2 weeks	14/03/2018	28/03/2018	System analyst
Development (room, booking, , report)	2 WEEK	29/03/2018	16/04/2018	Development team
Documentation Preparation	5 weeks	13/03/2018	17/04/2018	All members

TABLE 1.2 WORK BREAK DOWN

1.10.2 PROJECT ORGANIZATION

The project organization outlines the roles and responsibilities essential for designing, developing, testing, and deploying the hotel management system.

I. Project Manager

- ✓ Oversees project execution, timeline, and budget.
- ✓ Coordinates all team activities.
- ✓ Approves final deliverables.
- ✓ Ensures communication with the hotel owner and manager.

II. System Analyst

- ✓ Gathers and analyzes requirements.
- ✓ Designs system workflows (room management, booking, billing, reporting).
- ✓ Prepares use case diagrams, DFDs, ERD, and other models.
- ✓ Ensures requirement traceability throughout development.

III. Designer

- ✓ Designs user-friendly interface layouts for staff (receptionists, manager, cleaners).
- ✓ Ensures accessibility, readability, and ease of navigation.
- ✓ Conducts usability tests and refines interface based on feedback.

IV. <u>Software Developer</u>

- ✓ Implements all core modules:
- ✓ Room state management (clean, dirty, available, occupied)
- ✓ Booking and reservation (online, phone, walk-in)
- ✓ Billing and payment (cash, mobile banking, internet banking)

V. <u>Guest management</u>

- ✓ Automated daily reports (4:00 AM)
- ✓ Integrates modules, fixes bugs, and ensures system performance.
- ✓ Tester / Quality Assurance Specialist
- ✓ Performs unit, integration, and system testing.
- ✓ Validates booking rules, cancellation logic, and input accuracy.
- ✓ Conducts performance and stress testing.
- ✓ Collaborates with staff during UAT.
- ✓ Deployment & Training Team
- ✓ Installs the system on hotel devices.
- ✓ Sets up network connections.
- ✓ Trains all 17 hotel employees .
- ✓ Provides support during early operation.

1.10.3 TEAM ORGANIZATION

The project team consists of five members, each assigned specific roles for effective collaboration and successful system development.

1. Beimenet— Project Manager & System Analyst

- ➢ Leads planning, scheduling, and monitoring tasks.
- ➢ Confirms requirements with hotel management.
- ➢ Designs workflows for room, booking, payment, and reporting.
- ➢ Xxyrhw bbbbb byu5h[
- ➢ Collaborates on use cases, ERD, and DFD diagrams.

2. Sirawdink — System Analyst & Assistant Developer

- ➢ Assists in requirement gathering and documentation.
- ➢ Participates in modeling diagrams (Use Case, DFD, ERD).

- Supports development of booking and room modules.
- Helps create test scenarios with QA.

3. Fitsum— UI/UX DesignerDesigns interfaces for all user types.

- Creates visually clear and accessible layouts.
- Tests interface usability and applies improvements.

4. Abeni— Software Developer

- Develops core modules and business logic.
- Integrates booking, room, billing, payment, and reporting features.
- Performs debugging and ensures system reliability.

5. Merihun— Quality Assurance & Deployment Specialist

- Conducts all testing levels.
- Manages deployment and network setup.
- Leads training sessions for hotel staff.
- Performs stress testing to ensure system stability.

Communication Plan

- Weekly meetings to review project progress.
- Daily task updates within the team.
- Google Drive for shared documents and diagrams.
- Phone communication and telegram call for urgent issues.

1.11 BUDGET PLAN

The total estimated cost for developing and deploying the Shewaber Hotel Management System include various type of budget .This budget covers all phases of the project including analysis, design, development, testing,
deployment, documentation, training, and contingency expenses.

1. Development Costs

These costs cover the work required to design and build the hotel management system, including user interface design, system development, and quality assurance.

2. Deployment Costs

These are the expenses required to install and launch the system within the hotel Environment.

3. Operational Expenses

These cover the necessary support and upkeep for the system during its first year of operation.

4. Documentation and Printing

These expenses support project documentation preparation, printing of manuals, and related materials.

5. Contingency Budget

A reserved fund used for unexpected expenses such as extra storage needs, additional testing time, or small hardware replacements.

NO	Items	Quantity	Estimated cost
1	Computer And Pc	3	250,000
2	Pen	5	150
3	USB Cable	2	300
4	Flash Disk	1	2000
5	For Print	110 Page	2,500-3,000
6	Team Payment	5	155,000
TOTAL COST			409,550

TABLE 1.3 BUDGET PLAN

CHAPTER TWO : REQUIREMENT ANALYSIS AND SPECIFICATIONS

2.1 DESCRIPTION OF THE CURRENT SYSTEM

In present day, shewaber hotel relies on manual approach in regard to order management, reservation, billing and paying system which makes the process very inefficient, difficult to update, integrate and prone to error.

- ✓ Here is the picture of existing system

FIGURE 2.1 PICTURE OF EXISTING SYSTEM

2.2 PLAYERS/ACTORS IN THE EXISTING SYSTEM

- Manager-overseeing all hotel operation and inventory.
- ordering customer places orders.
- Reception-checking, confirming and modifying reservation according to customer needs.
- Administrator-overseeing all aspects of the hotel daily operations through management system.

2.3 USE CASE DIAGRAM FOR EXISTING SYSTEM

This system is entirely manual, as evident from attached use case diagram, that depicts interaction with customers/staff/ managers, using papers and pen, etc.

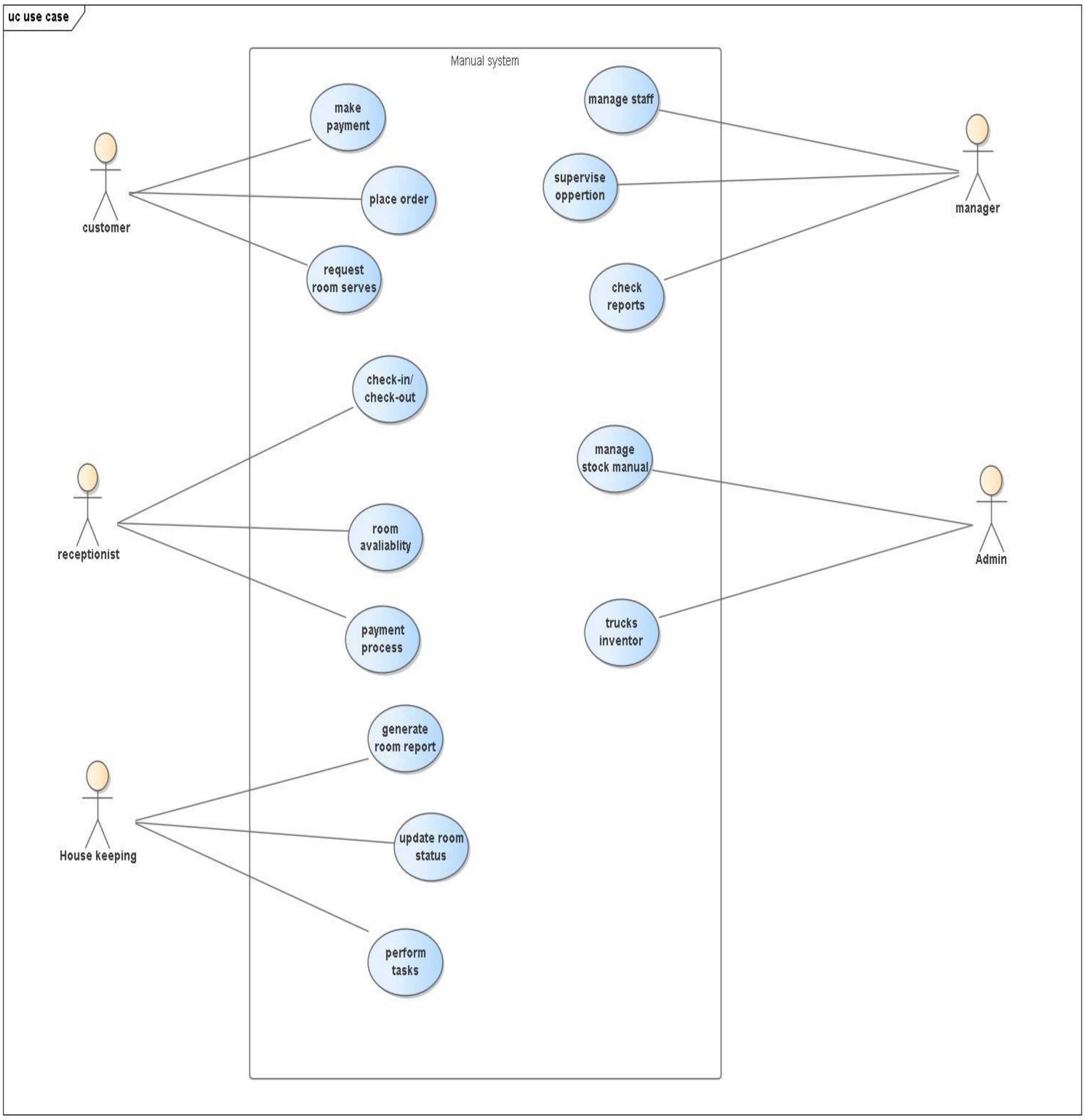


FIGURE 2.2 USE CASE DIAGRAM FOR EXISTING SYSTEM

2.4.FORM/DOCUMENTS USED IN THE EXISTING SYSTEM

The current manual system uses paperworks to manage the hotel. Booking is done through phone calls or through visit to the hotel booking office. The guest's personal details such as Name, Age, Nationality, and Duration of stay, are input during booking in. This reliance on memory can lead to potential errors and inefficiencies in order to manage.

2.5 BUSINESS RULE FOR EXISTING SYSTEM

The Business Rules of the **Shewaber Hotel** are the established procedures and policies that currently govern operations within the manual, paper-based environment. These **rules,** while ensuring basic integrity, are the core source of the system's current **slowness.**

R1➛Room Assignment Protocol

A room is assigned only after a staff member, typically the Reception Cleaner, physically verifies the room's status in the ledger. This sequential, manual check is the hotel's existing safeguard against double-booking.

R2➛Mandatory Booking Data

A reservation is officially recorded and recognized only when staff capture a complete set of critical information: the Guest Name, the unique Identity Document Number (Passport, ID, or License), the Dates of stay, and the Payment Receipt ID. This rule is essential for regulatory compliance and tracing financial proof.

R3➛Payment Processing

Transactions must be logged using one of the three accepted methods: Mobile Banking**,** Internet Banking, or Direct Cash**.** The system must respect and track these specific financial channels.

R4➛Transaction Security (Cancellation/Changes)

To authorize any change to a booking **(modification or cancellation)**, the customer must provide the Payment Receipt Number or the original Booking Number. This policy acts as a manual security check to prevent fraudulent alterations.

R5➛Daily Reporting Schedule

Management demands that all key operational and financial summaries—the Room Status, Revenue, and Guest Information reports—be manually compiled and ready for review every day at **4:00 AM**

2.6 PROPOSED SYSTEM

2.6.1 OVERVIEW OF THE NEW SYSTEM

The new system will automate core functions, including booking, billing and report generation. It will also optimize operations by enabling features such as reservation management, staff management, and performance tracking, ensuring efficiency and improved service delivery.

2.6.2 BUSINESS RULE OF THE NEW SYSTEM

- The system automatically processes bookings, updates room availability in real-time and sends confirmations.
- Payments can be accepted via multiple methods. *The system updates the bill automatically when aservice is add
- A room cannot be double booked for the same date *unauthorized access should be blocked

2.6.3 FUNCTIONAL REQUIREMENTS

- The Proposed System is the computer-based Hotel Management System (HMS) designed to replace the current paper ledger. Its development is the primary goal to transform the hotel into a provider of modern hospitality.

Here is a list of some functional requirements

I. Ensure secure login and logout functionality.
II. Implement role based access for admin, staff, and other users.
III. Accept digital payments method.
IV. Generate accurate bills based on using services.
V. Provide allow reservation cancellation or modification.
VI. Must store all payments in database.
VII. Generate daily, weekly, monthly reports.
VIII. Generate store gests personal information.
IX. provide record customer feedback .
X. provide record check in/check out date and time.

1. Room Management Module

Room Detail Storage: The system must record and maintain all physical room attributes, including Room Number, Room Type, Floor, and the number of Beds.

Room Status Tracking: The system must accurately reflect and transition between the five **operational statuses:** Available, Occupied, Reserve, Check-out, and Pending (for cleaning). This is crucial for efficient assignments.

Check in/Assignment: The system must allow staff to allocate a room to a guest and immediately change the room's status to Occupied, ensuring instant inventory updates.

2. Booking and Reservation Module

Guest information : The system must include forms that require the input of ***Guest*** **Name, ID/Passport/License Number, Dates, and Payment Receipt ID for every reservation, securing data integrity.**

Rapid Availability Search: Staff must be able to query the system to quickly determine which rooms are available based on user defined criteria (Date Range and Room Type).

Modification and Cancellation: The system must process room changes or cancellations, but only after validating the request using the Payment Receipt Number or Booking Number and once payment complete customer can’t cancel booking and reservation.

3. Payment Module

Payment Method Logging: The system must accurately record the specific method used for payment: **Mobile Banking or Internet Banking.**

Receipt Generation: The system must automatically generate a traceable, unique Payment Receipt ID for every financial transaction.

4. Reporting Module

On-Demand Report Generation: The system must be capable of generating reports covering the three required categories: **Room Occupancy/Availability, Revenue Summaries, and Guest Information Lists.**

Or **("how often gust can stay ?"➛daily, weekly, monthly).**

Automated Scheduling: The system must be programmed to automatically compile and save all necessary reports every day at **4:00 AM**, fully automating the critical managerial deadline.

2.6.4.NON-FUNCTIONAL REQUIREMENTS

A. Performance Requirements

The system must respond to common operations such as room search, booking, check-in, and check-out within fast time. The system Must support multiple staff users working simultaneously without slowing down.

Report generation (daily, weekly, monthly) must complete within maximum 2:00 minute .

B. Reliability Requirements

The system must operate continuously with minimal downtime during hotel business hours.

Must safely handle unexpected shutdowns or power failures without data loss.

Should support local caching so main tasks (check-in/out, room status updates) can continue during short network interruptions.

C. Availability Requirements

The system should be available most of time , except during scheduled maintenance.

Maintenance activities should be planned so they do not interrupt core hotel operations such as booking and check-in.

D. Security Requirements

All users must log in using a unique username and password.Sensitive processes such as refunds, reservation cancellations, and room price changes require role-based authorization.

Guest personal data and payment details must be encrypted during storage and transfer.

The system must record audit logs of all important actions (booking, payment, cancellation, room status change).

E. Usability Requirements

The system must be simple, clean, and easy to navigate for staff with basic computer skills.

New employees should be able to learn the system with short training sessions.

Interface must use clear labels, readable text, and consistent design.

F. Scalability Requirements

The system must support future expansion, including:

More rooms

New services

Additional hotel branches

Database must handle increased data without requiring major redesign.

G. Maintainability Requirements

The software should be modular, making updates and improvements easy.

All components must be well-documented so future technicians can maintain or extend the system.

System should allow smooth updates without affecting stored data.

H. Data Accuracy Requirements

The system must validate input data (dates, ID numbers, phone numbers, payment references).

Should automatically detect and prevent duplicate reservations or duplicate guest profiles.

Calculations (pricing, taxes, discounts) must be accurate and consistent.

I. Disaster Recovery Requirements

Automatic daily backups must run to protect data.

Weekly full backups and hourly incremental backups (optional, recommended).

System must have a simple restore process in case of database damage.

J. Compatibility Requirements

Must function on commonly used web browsers (Chrome, Edge).

Should support local printers for bills, invoices, and reports.

Should be compatible with standard hotel POS devices.

K. Legal and Compliance Requirements

The system must protect guest information according to local data-protection rules.

Financial records must be stored properly for audit and legal requirements.

L. Accessibility Requirements

Interface must use readable fonts, good color contrast, and simple layout.

Should be easy to use for staff with basic technology skills.

2.6.5.ACTOR AND USE CASE IDENTIFICATION

1. Actors

- **hotel Manager** : reviews reports, approves overrides.
- **Payment gateway** : accept online payment from customer .
- **System Administrator** :manages system settings and user accounts.
- **Customer** :rooms and makes reservations.

2. Use Cases

- Login / Logout
- Search room
- Make payment
- Receive confirmation
- Manage staff
- View report
- Set pricing
- Manage inventory
- Validate payment
- Reject payment
- Process payment
- View system logs
- Manage system

- Manage account

2.6.6 SYSTEM MODEL

2.6.6.1 USE CASE MODEL

Actors and use cases:

Customers :

- search room
- make payment
- receive confirmation

Administrator :

- manage user account
- manage system setting
- view system Logs

Manager :

- manage staff
- view report
- set pricing
- manage inventory

payment :

- process payment
- validate payment
- reject payment

- Use case diagram for shewaber hotel Room reservation system

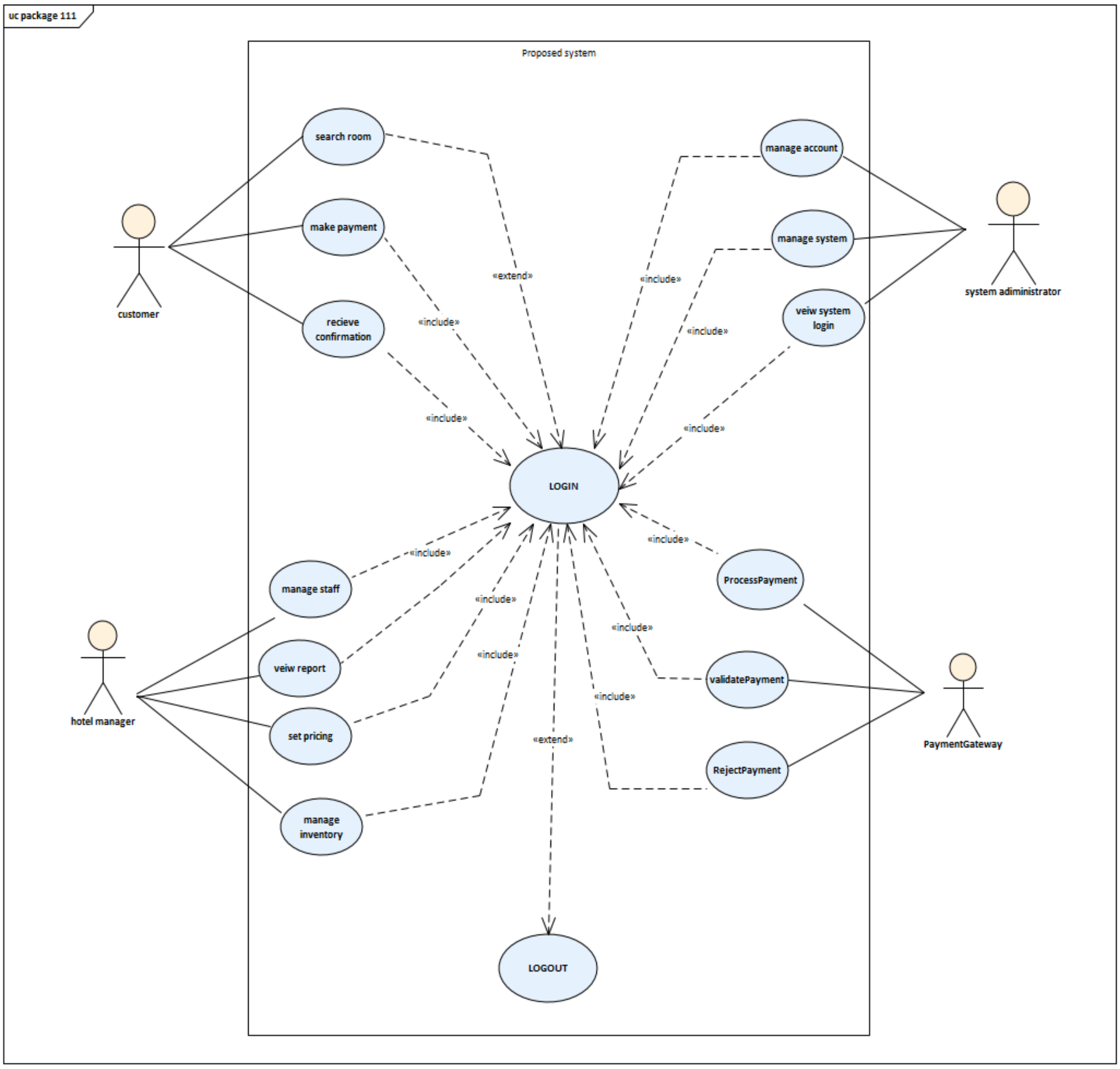


FIGURE 2.3 USE CASE FOR PROPOSED SYSTEM

2.6.6.1.1 USE CASE DESCRIPTION

Use case	Actor	Description	precondition	Basic flow	Post condition
Search room	Customer	Search for available rooms based on criteria	Customer is browsing hotel website	1.selecct search parameter 2.system display available room 3.filter result	Customer view room option
Make payment	CUSTOMER	Make a payment for booking service	Customer has pending bill /reservation	1.select payment method 2.enter payment detail 3.system process payment 4.receve receipt	Payment recorded ,bill update
Receive confirmation	Customer	Receive booking /payment confirmation	Customer complete booking /payment	1.system generate confirmation 2. Customer view confirmation	Confirmation delivered to Customer
Manage inventory	Manager	Track and manage hotel supplies	User has inventory management permission	1.view current inventory levels 2.update stock after usage /receiving 3.gennerate low stock alert 4.place order	Inventory record alert
Set pricing	Manager	Define room rates and service charge	User ha pricing authority	1.access pricing module 2.set base rate , seasonal pricing 3.define discount 4.apply charges	New pricing active in system

View report	Manager	Generate and analyze hotel performance report	User has reporting permission	1.select report type 2.define data range 3.generate report 4.analize data	Report generate for decision making
Manage staff	Manager	Handle staff scheduling ,role ,and information	User has staff management privilege	1.view staff roster 2.assing shift /role 3.update staff information 4.manage permission	Staff records & schedules update
Process payment	Payment	Handle payment transaction	Booking bill has been generated	1.select payment method 2.process transaction 3.record payment 4.issue receipt	Payment recorded and update
Validate payment	Payment	Check if payment is correct or not .	Selected payment methods	1.Customer enter payment detail 2.system sends payment 3payment gateway check 4.Approved or declined payment	Room booking is confirmed
Reject payment	Payment	Check the payment cannot be processed successfully	Select payment methods	1.customer enter payment detail 2.system sends payment 3.payment getway check 4.declined payment	Show Reject Message

Manage system	Administrator	Configure and maintain hotel management system	User has full system access	1.configure system settings 2.manage integration 3.perform system update 4.back up /restore data	System properly configured and maintained
View system logs	Administrator	Control a system activity and audit trails	User log viewing permission	1.access log management module 2.filter log by criteria 3.review system events 4.export logs	System activity monitored & recorded
Manage account	Administrator	Overall system administration and management	User has administrative privileges	1.overse all system function 2.manage user role/ Permissions 3.handle system wide configurations 4.resolve technical issues	Administration tasks completed

TABLE 2.1 USE CASE DESCRIPTION

I. SEARCH ROOM

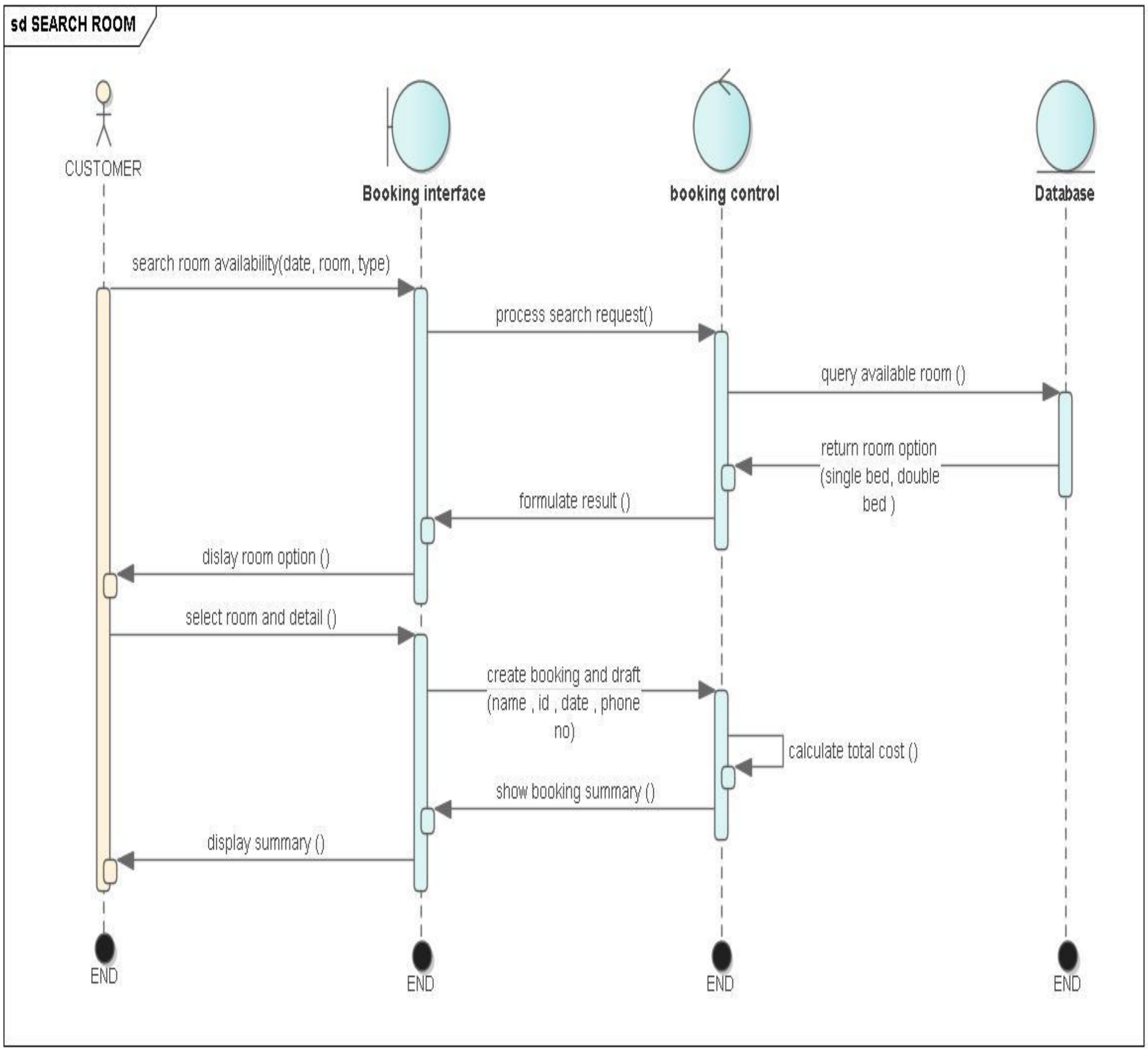


FIGURE2.4 SEQUENCE DIAGRAM SEARCH ROOM

II. MAKE PAYMENT

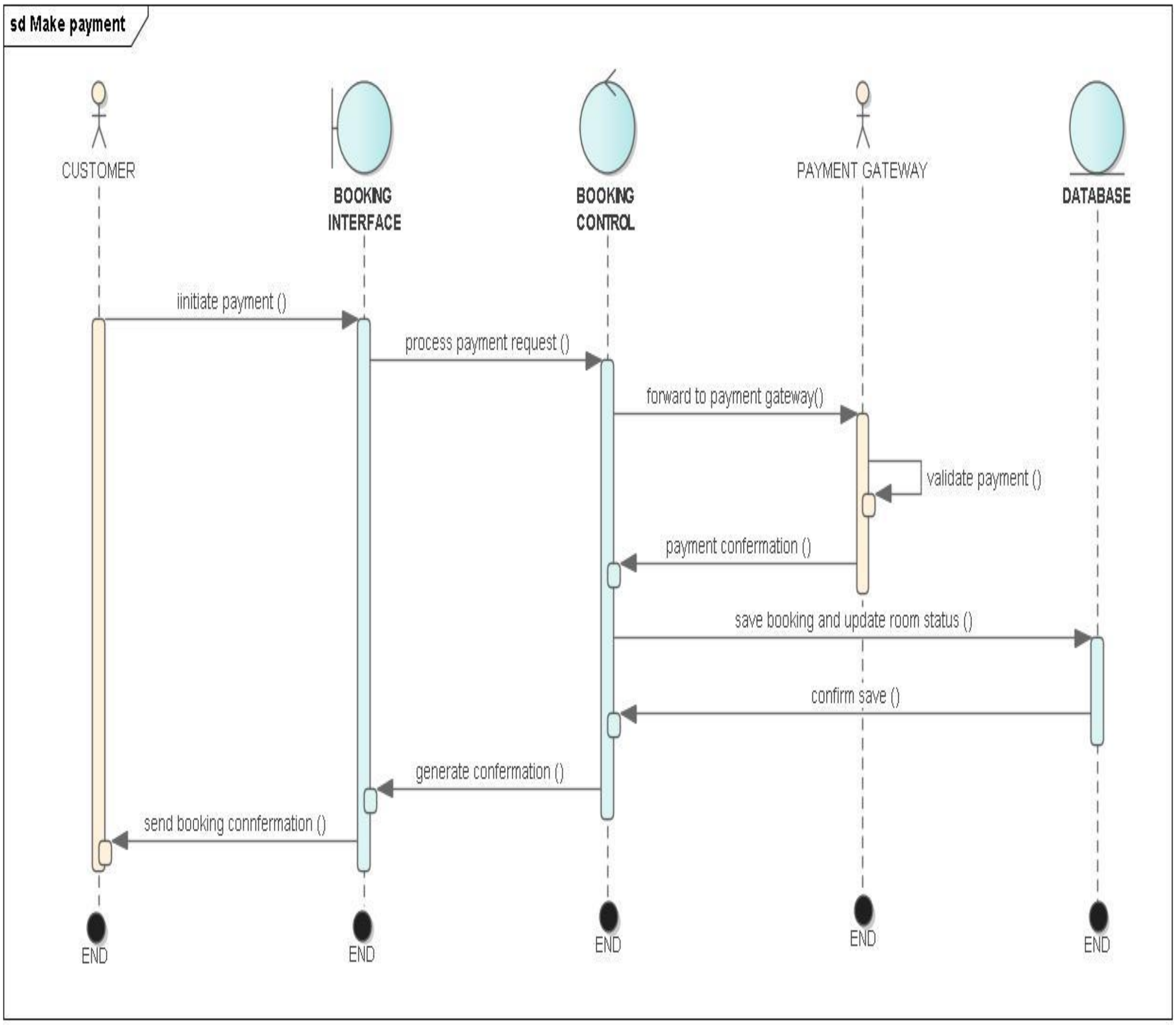


FIGURE2.5 SEQUENCE DIAGRAM MAKE PAYMENT

III. RECEIVE CONFIRMATION

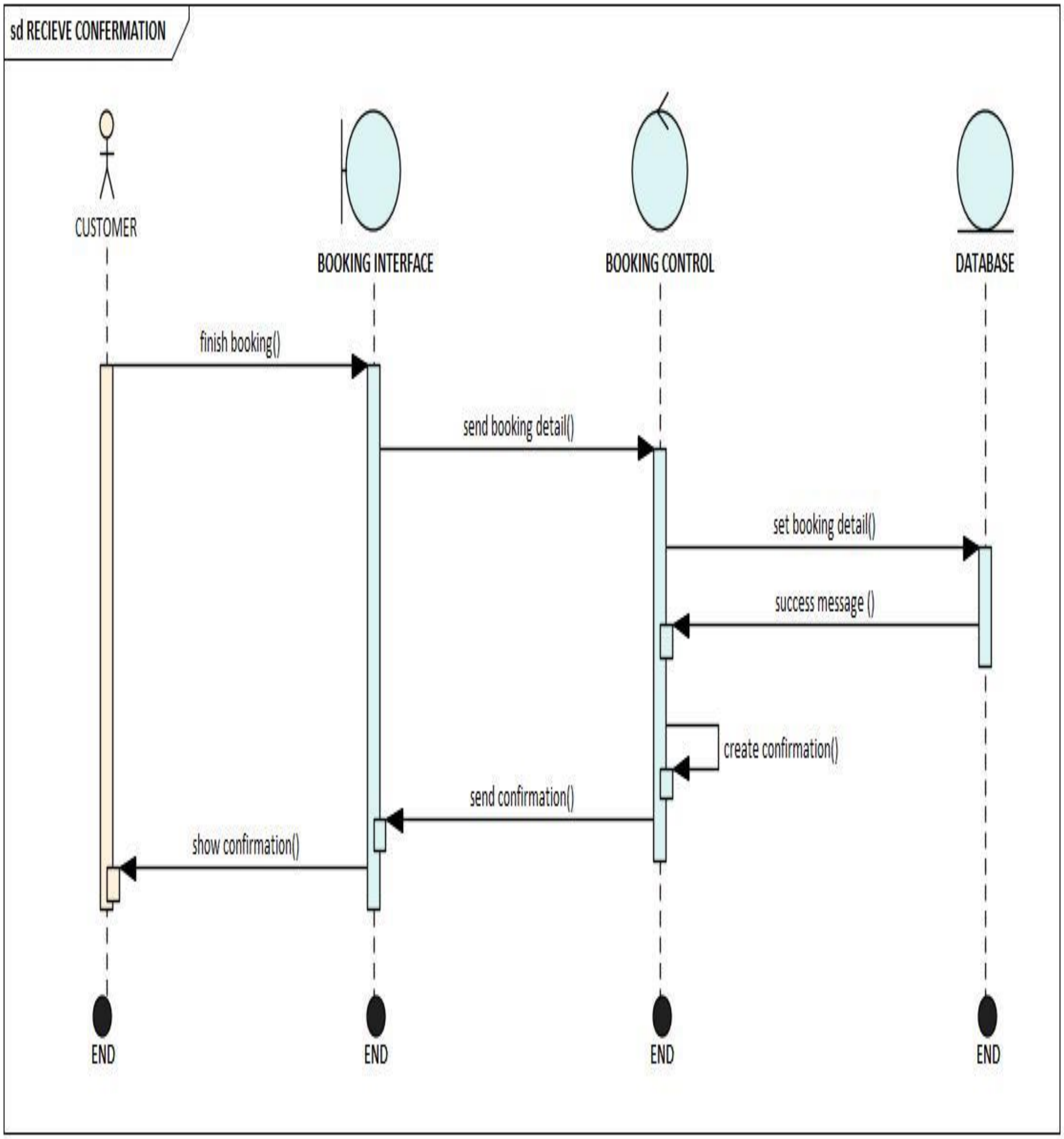
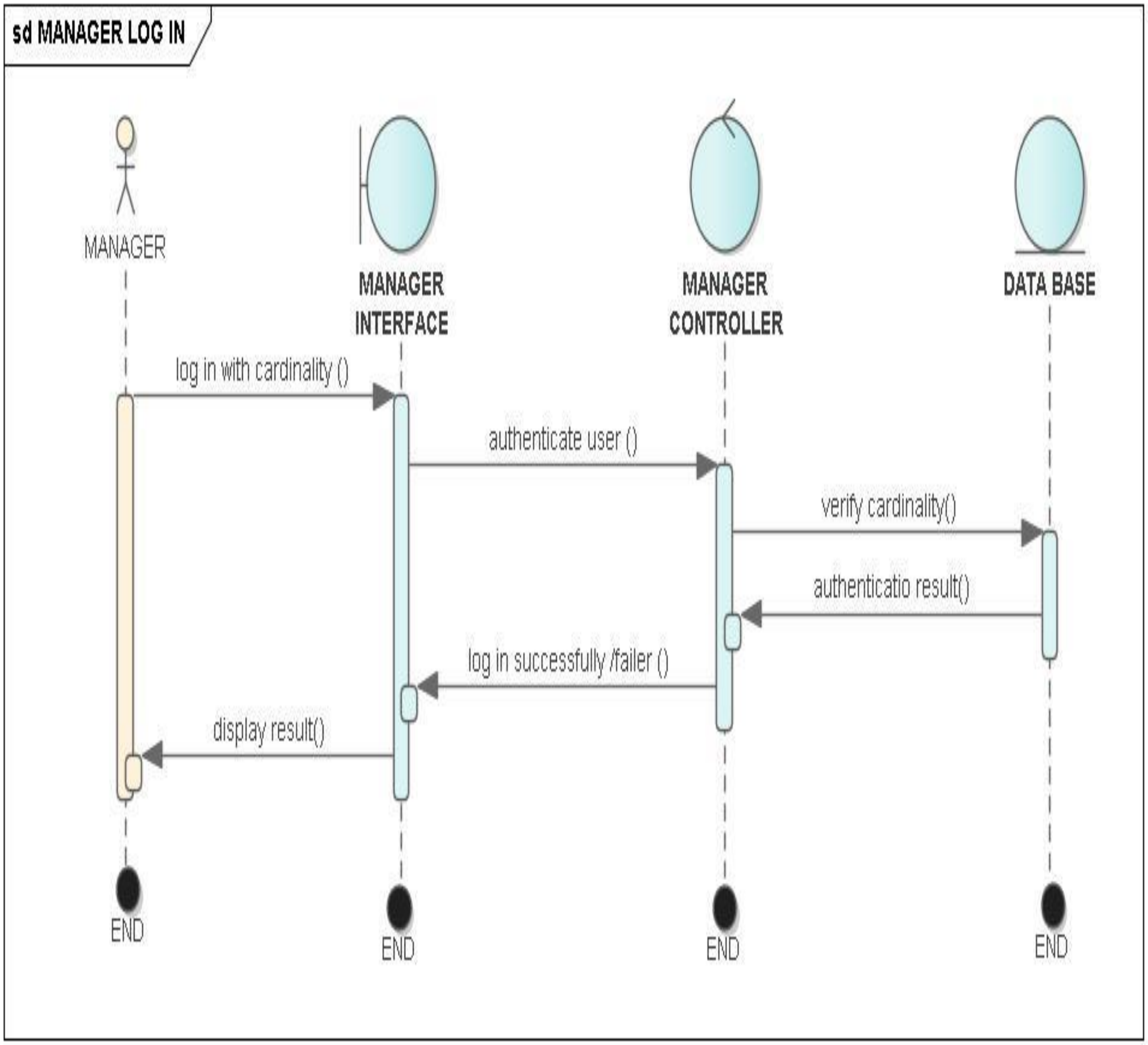


FIGURE 2.6 SEQUENCE DIAGRAM RECEIVE CONFIRMATION

IV. MANAGER LOG IN



.

FIGURE2.7 SEQUENCE DIAGRAM MANAGER LOG IN

V. MANAGE INVENTORY

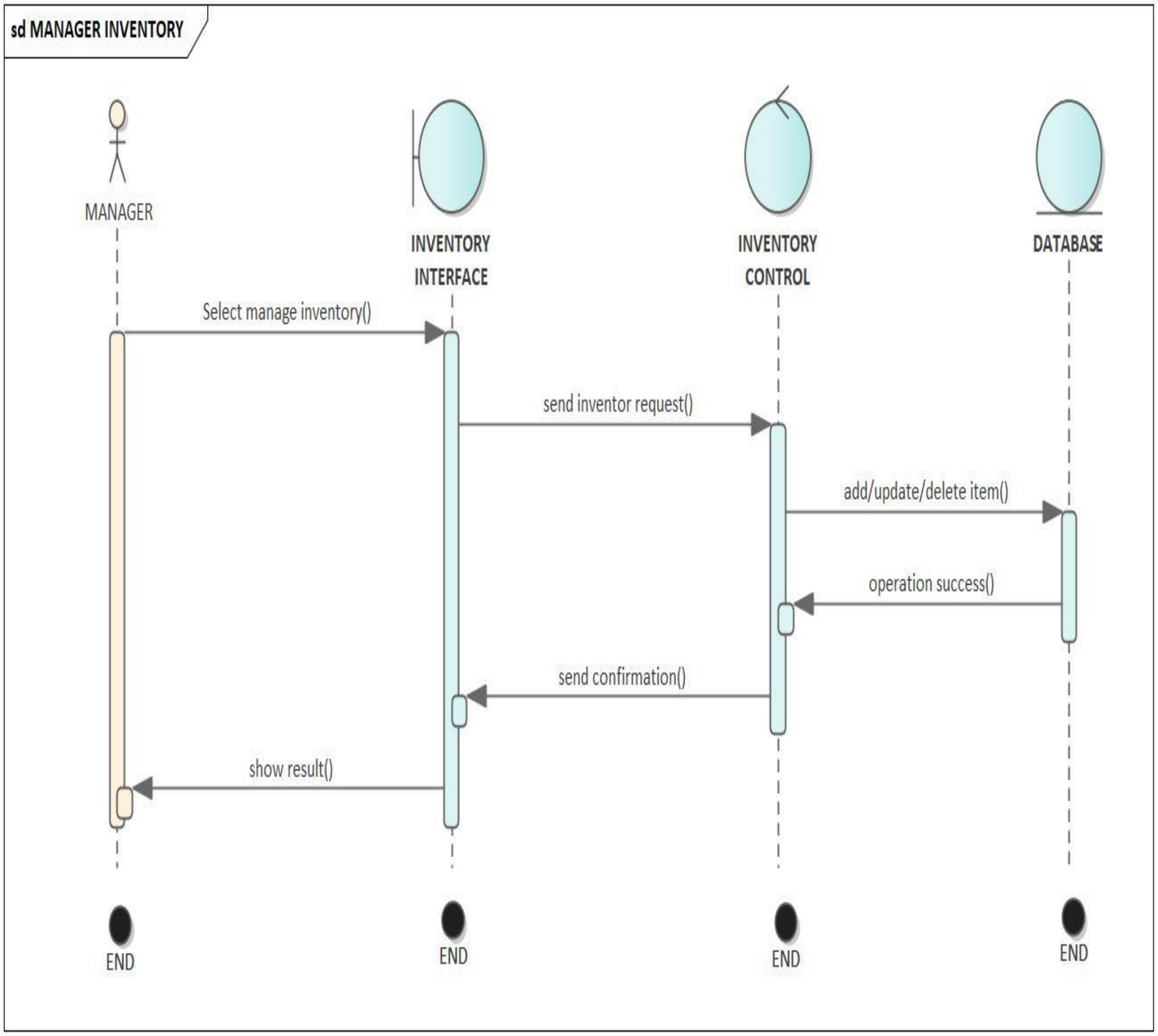


FIGURE 2.8 SEQUENCE DIAGRAM MANAGE INVENTORY

VI. VIEW REPORT

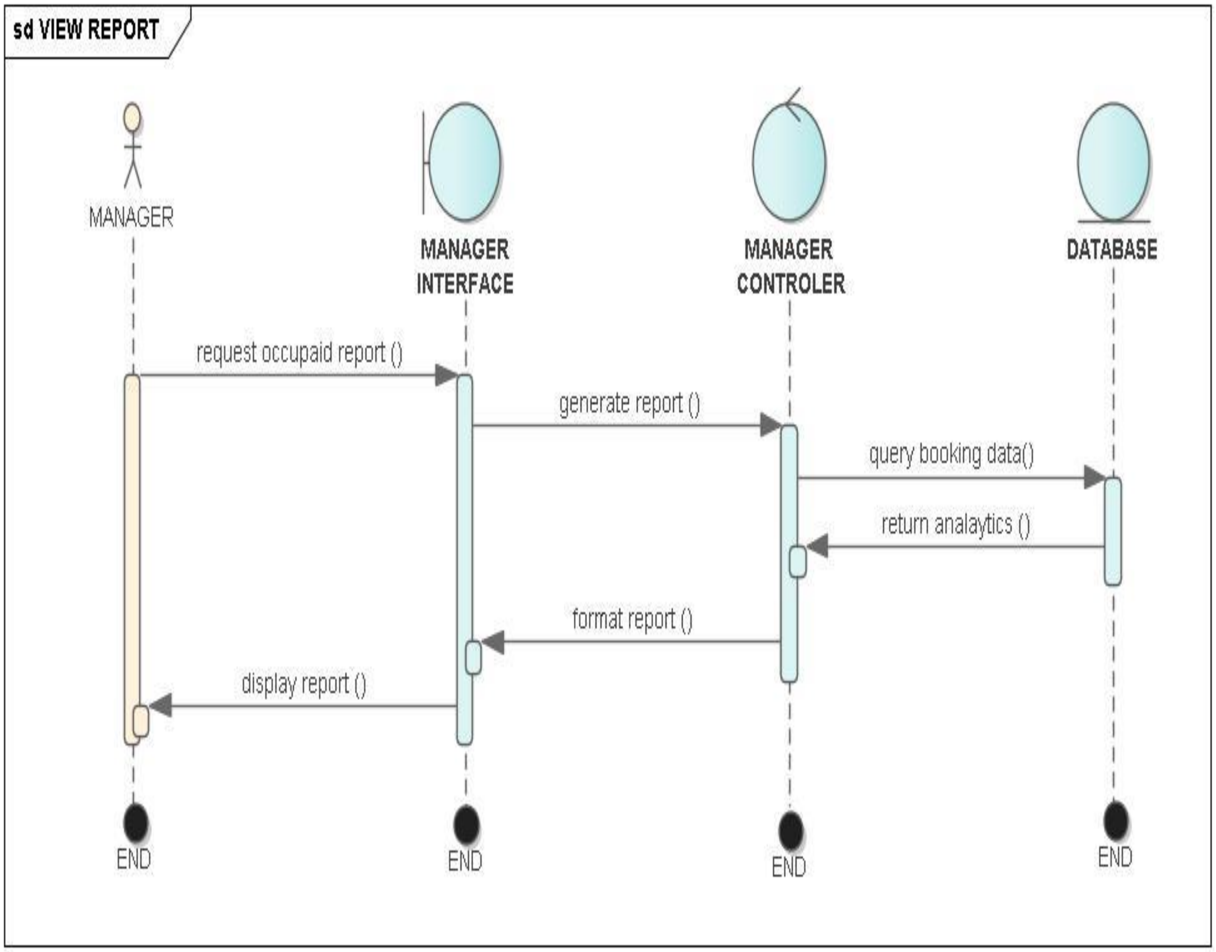


FIGURE2.9 SEQUENCE VIEW REPORT

VII. SET PRICING

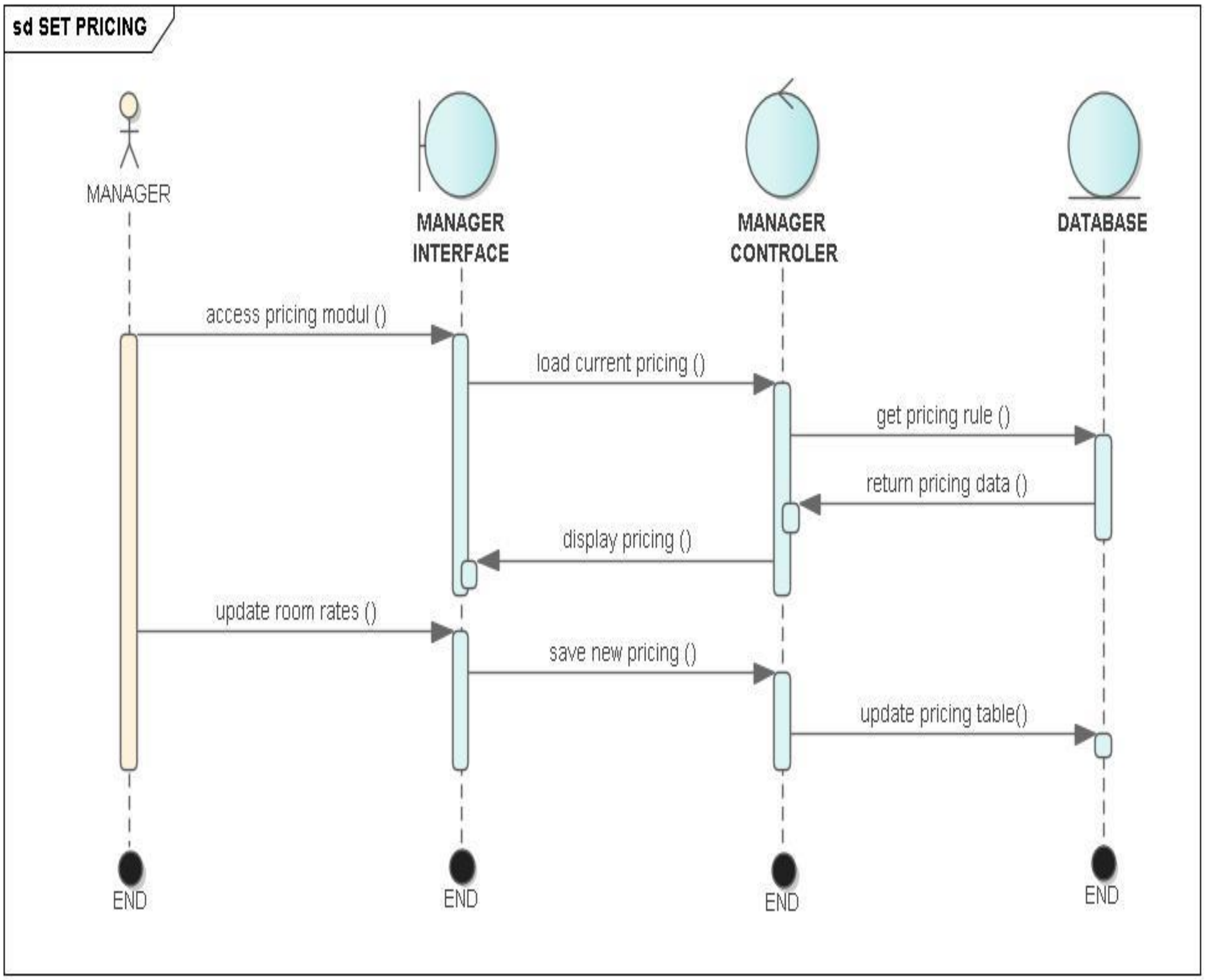


FIGURE2.10 SEQUENCE DIAGRAM SET PRICING

VIII. PAYMENT REJECTION

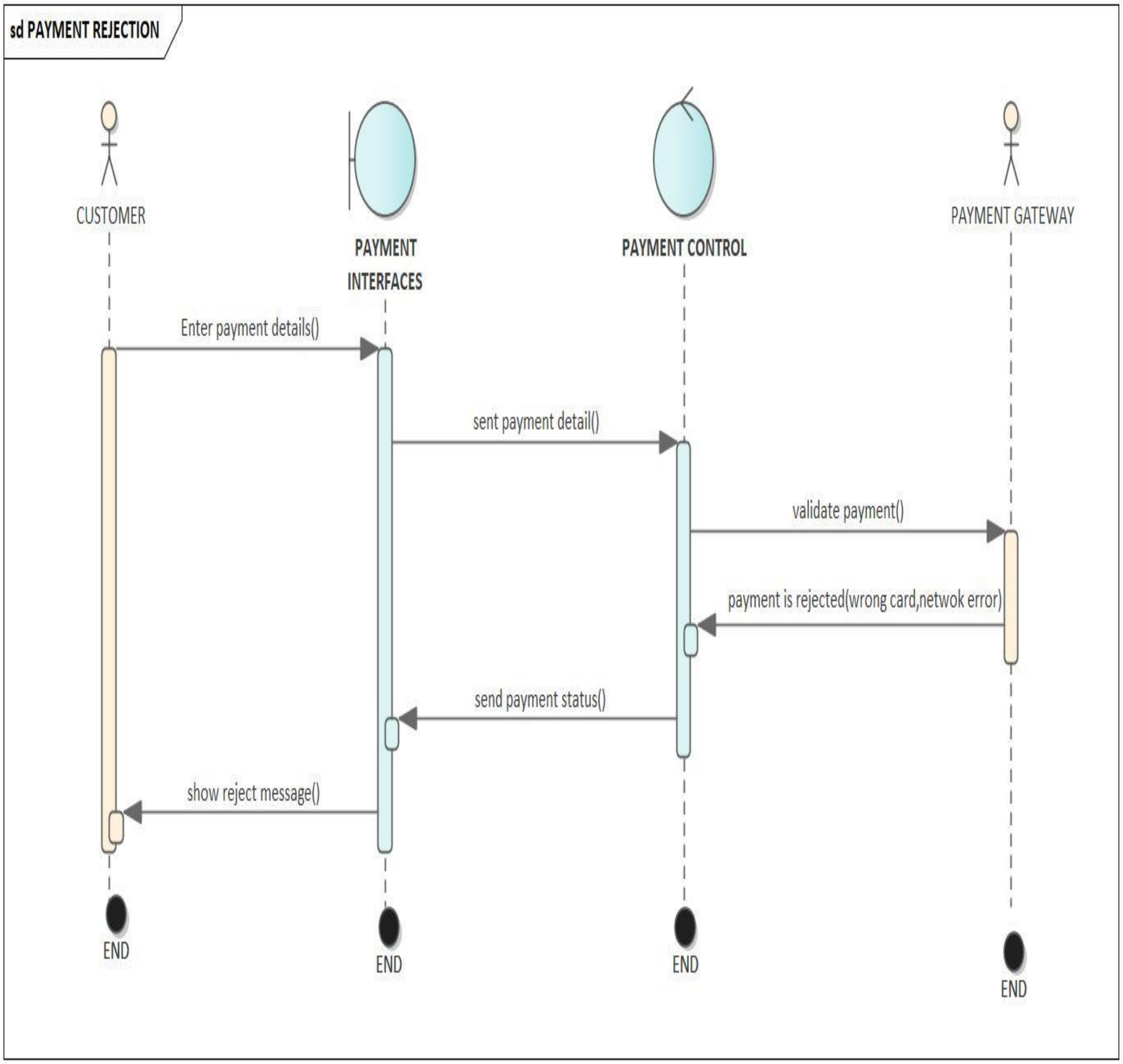


FIGURE2.11 SEQUENCE DIAGRAM PAYMENT REJECTION

IX. SYSTEM ADMIN LOG IN

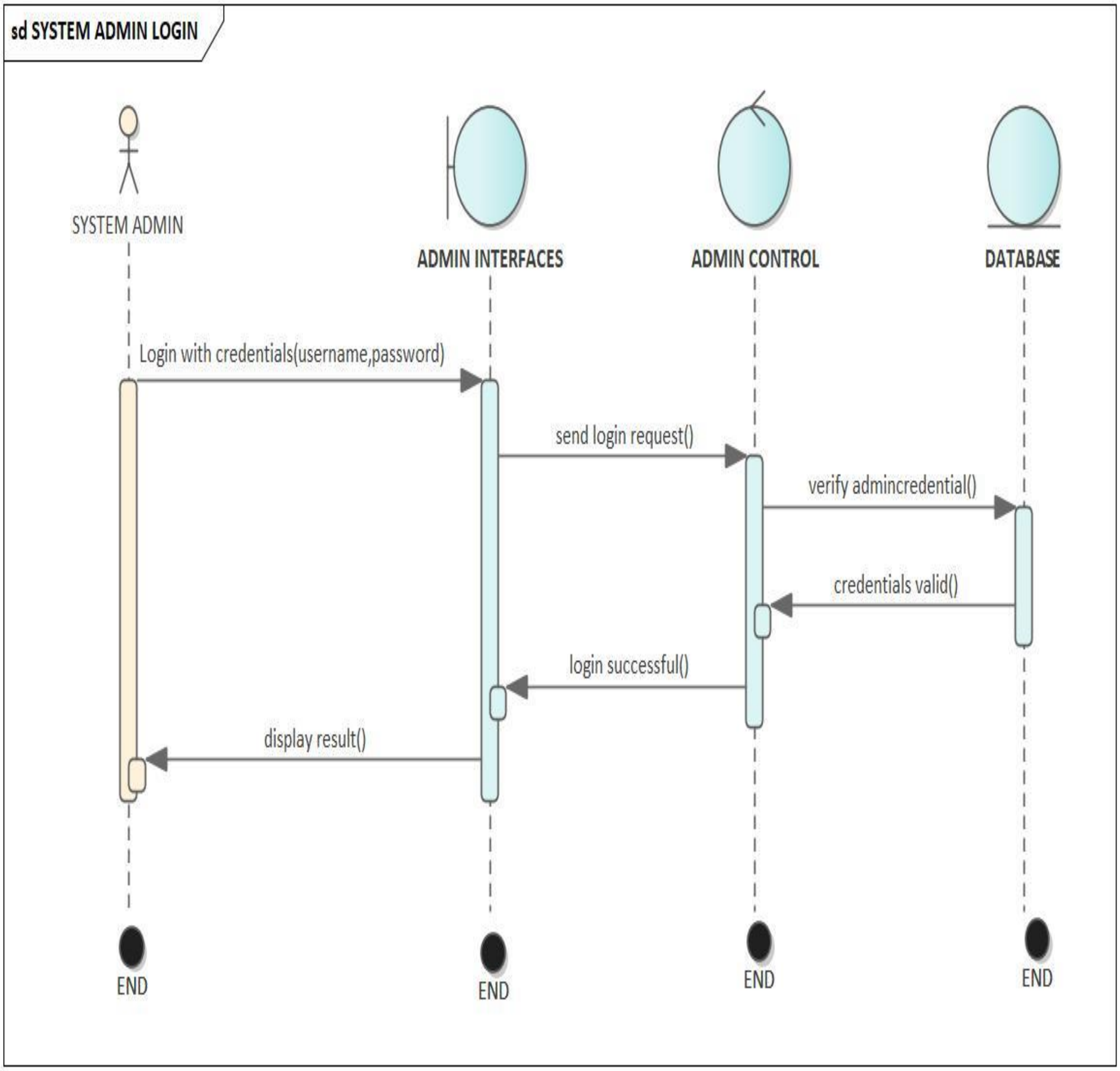


FIGURE2.12 SEQUENCE DIAGRAM SYSTEM ADMIN LOG IN

X. MANAGE ACCOUNT

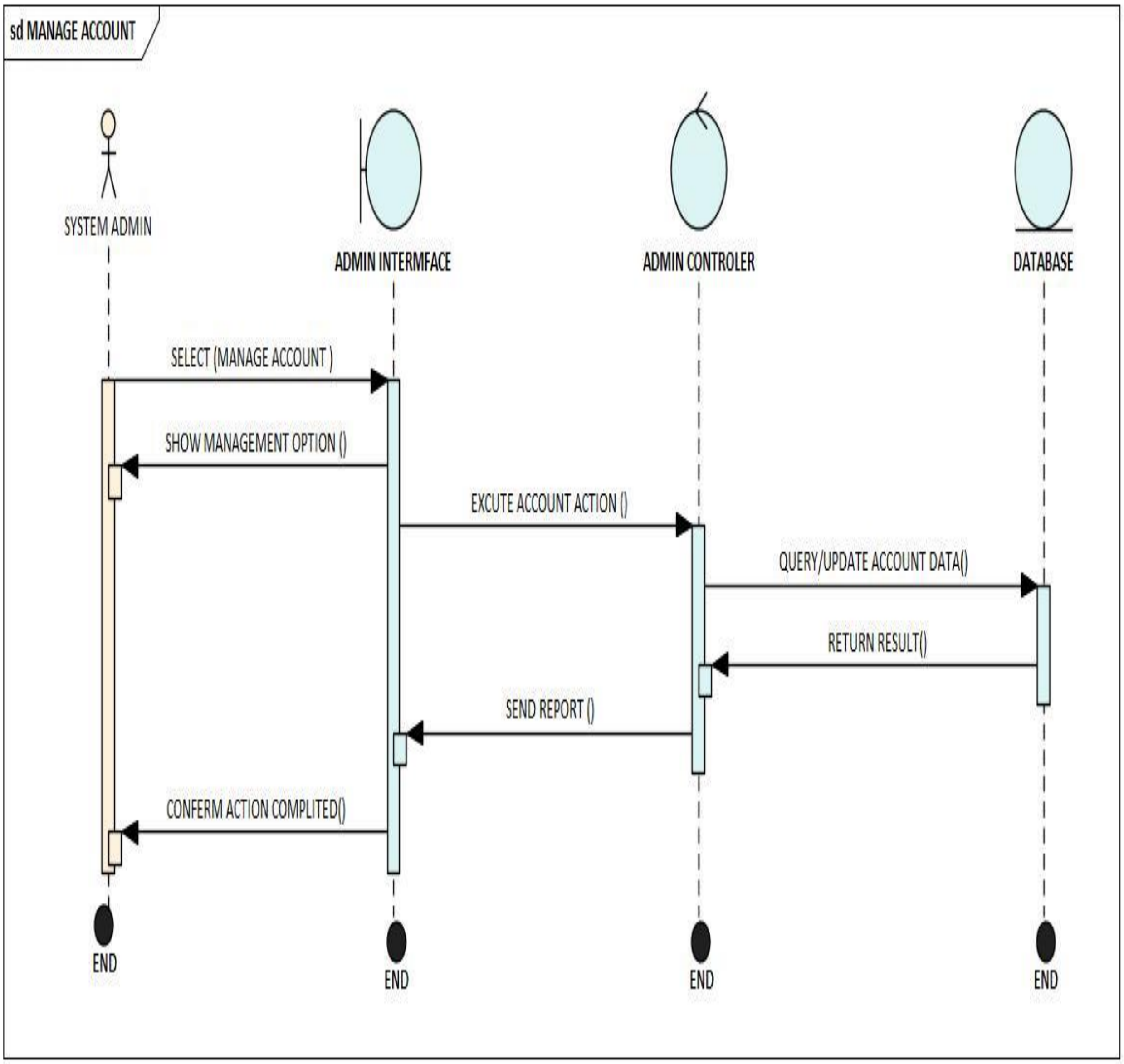


FIGURE2.13 SEQUENCE DIAGRAM MANAGE ACCOUNT

2.6.6.3 ACTIVITY DIAGRAM

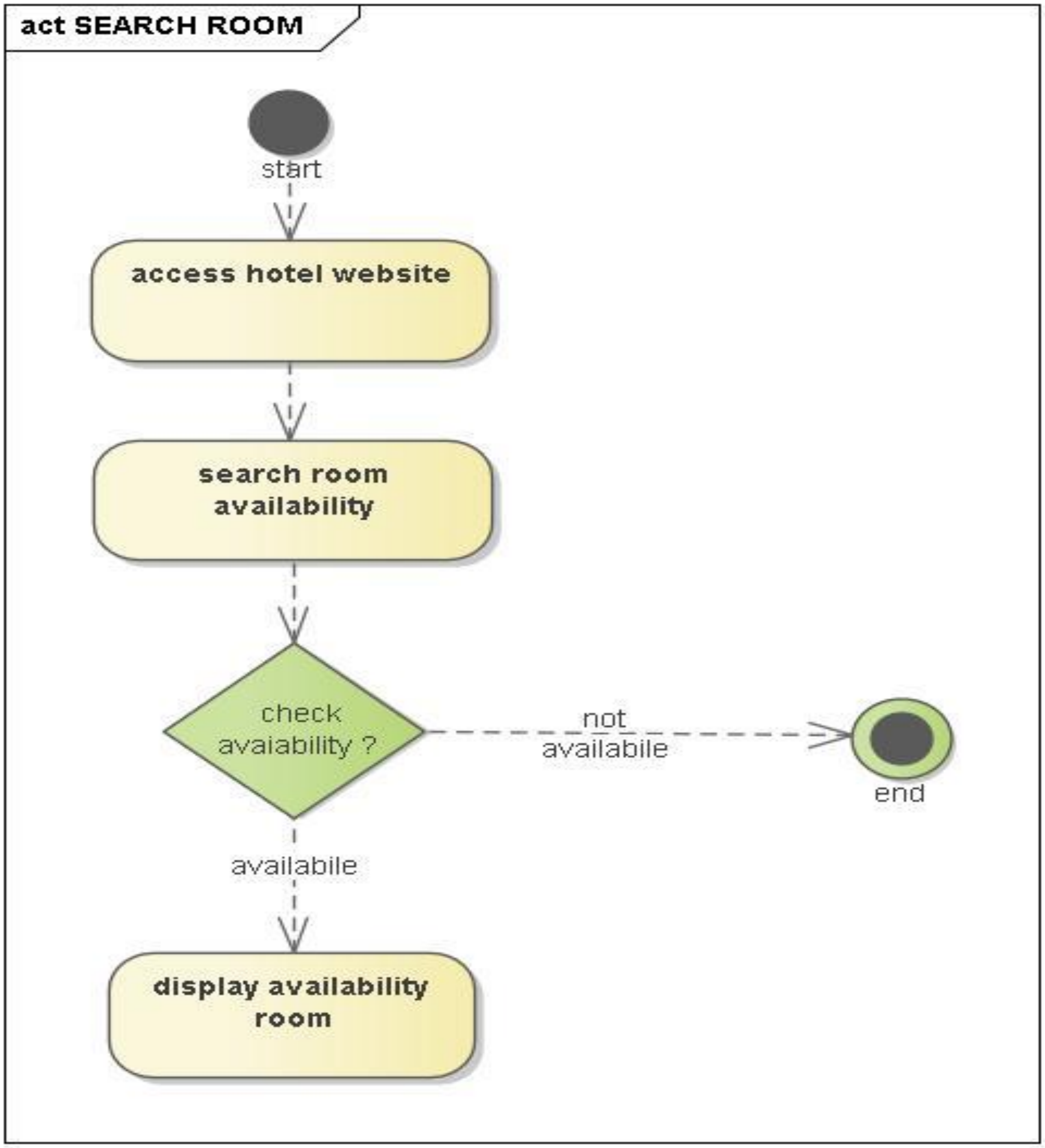


FIGURE2.14 ACTIVITY DIAGRAM SEARCH ROOM

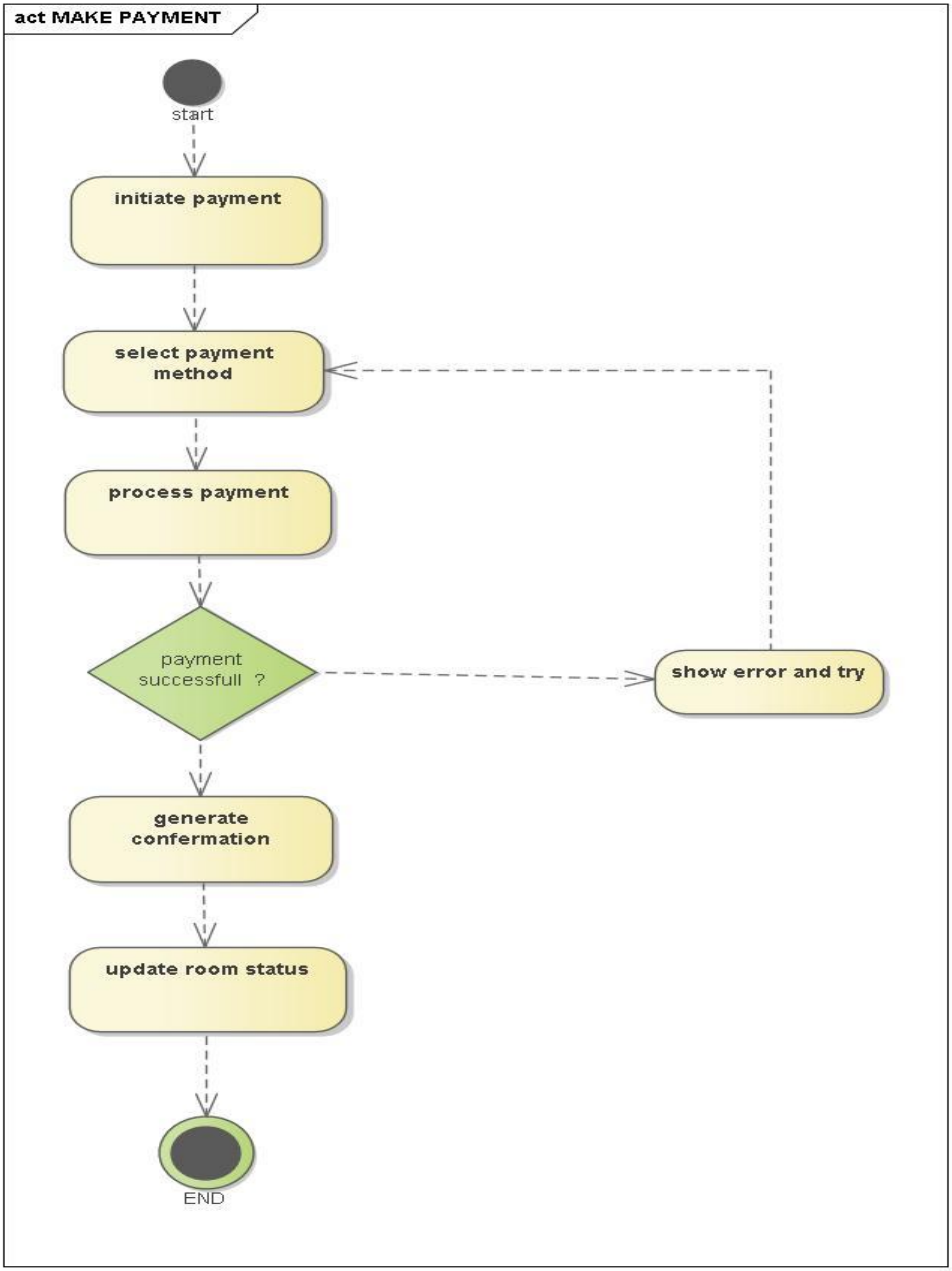


FIGURE2.15 ACTIVITY DIAGRAM MAKE PAYMENT

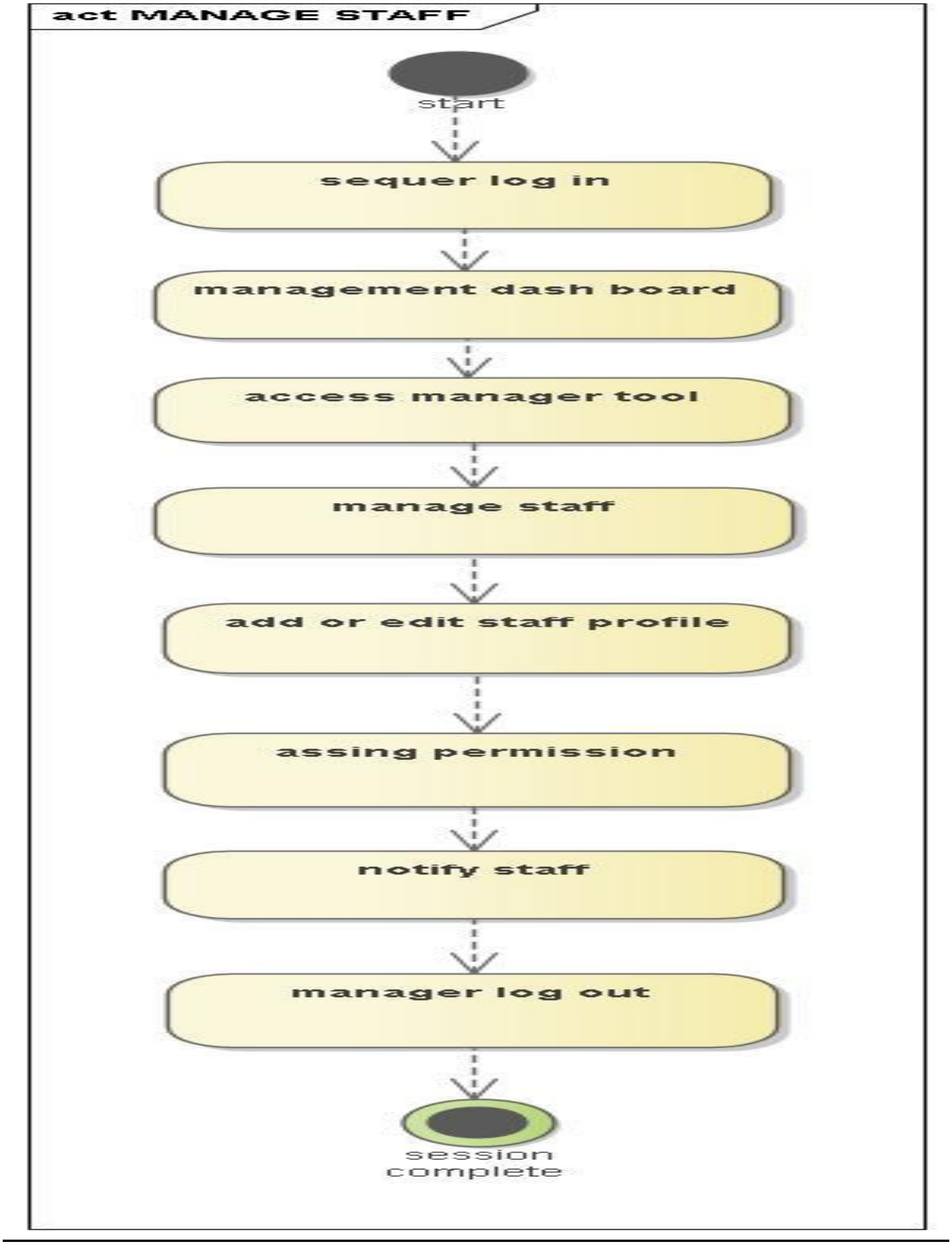


FIGURE2.16ACTIVITY DIAGRAM MANAGE STAFF

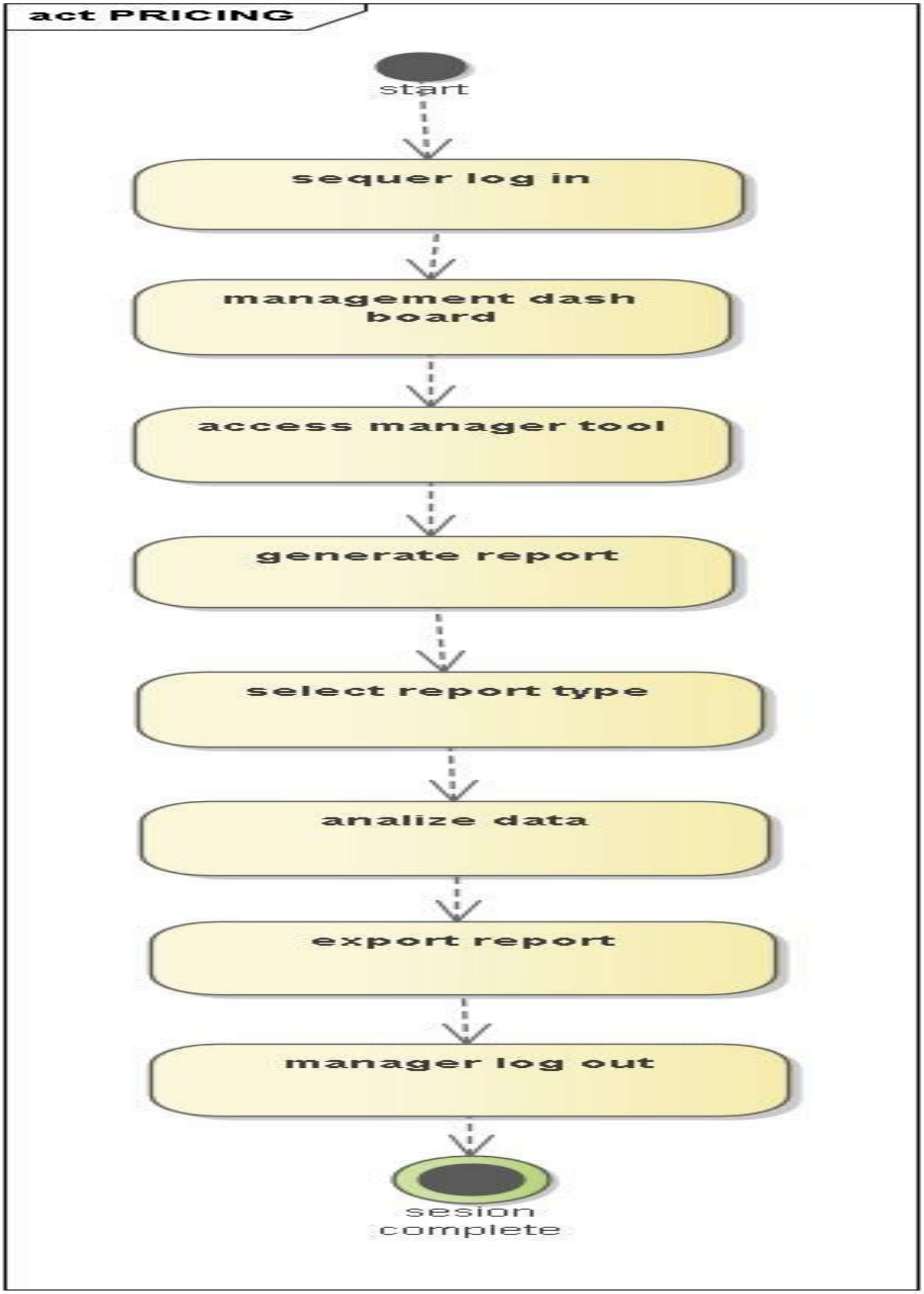


FIGURE2.17ACTIVITY DIAGRAM PRICING

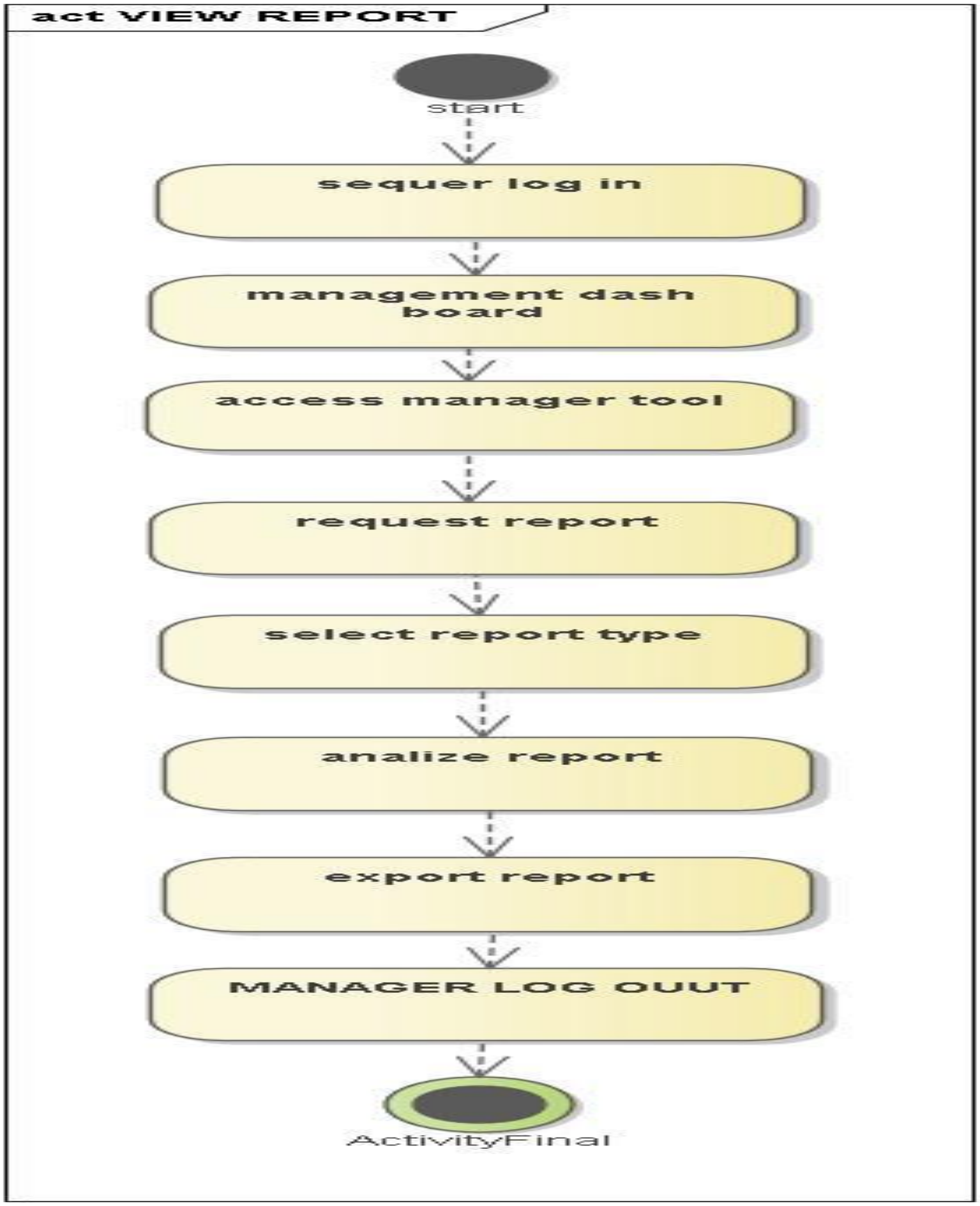


FIGURE2.18ACTIVITY DIAGRAM VIEW REPORT

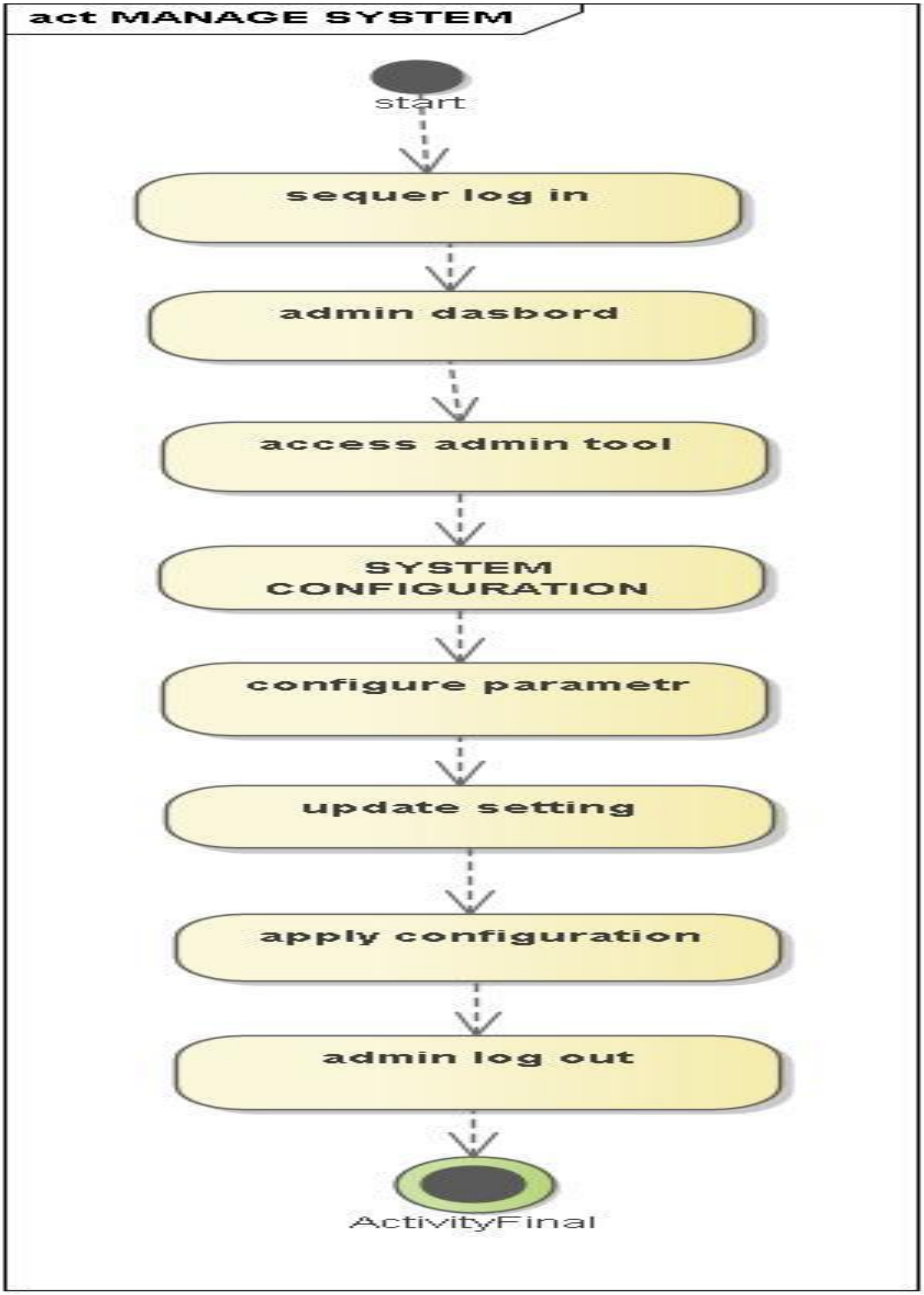


FIGURE2.19 ACTIVITY DIAGRAM MANAGE SYSTEM

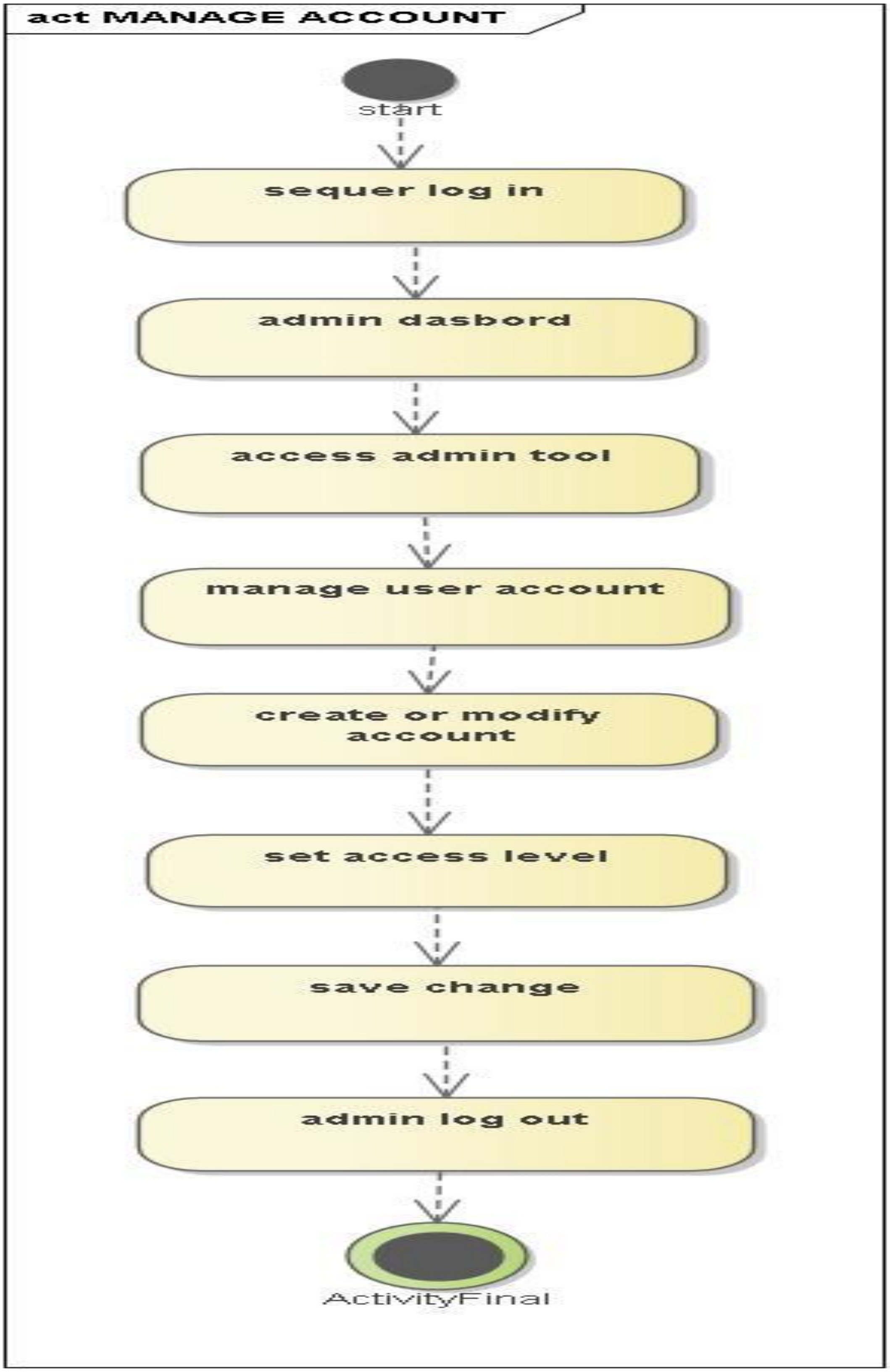


FIGURE2.20 ACTIVITY DIAGRAM MANAGE ACCOUNT

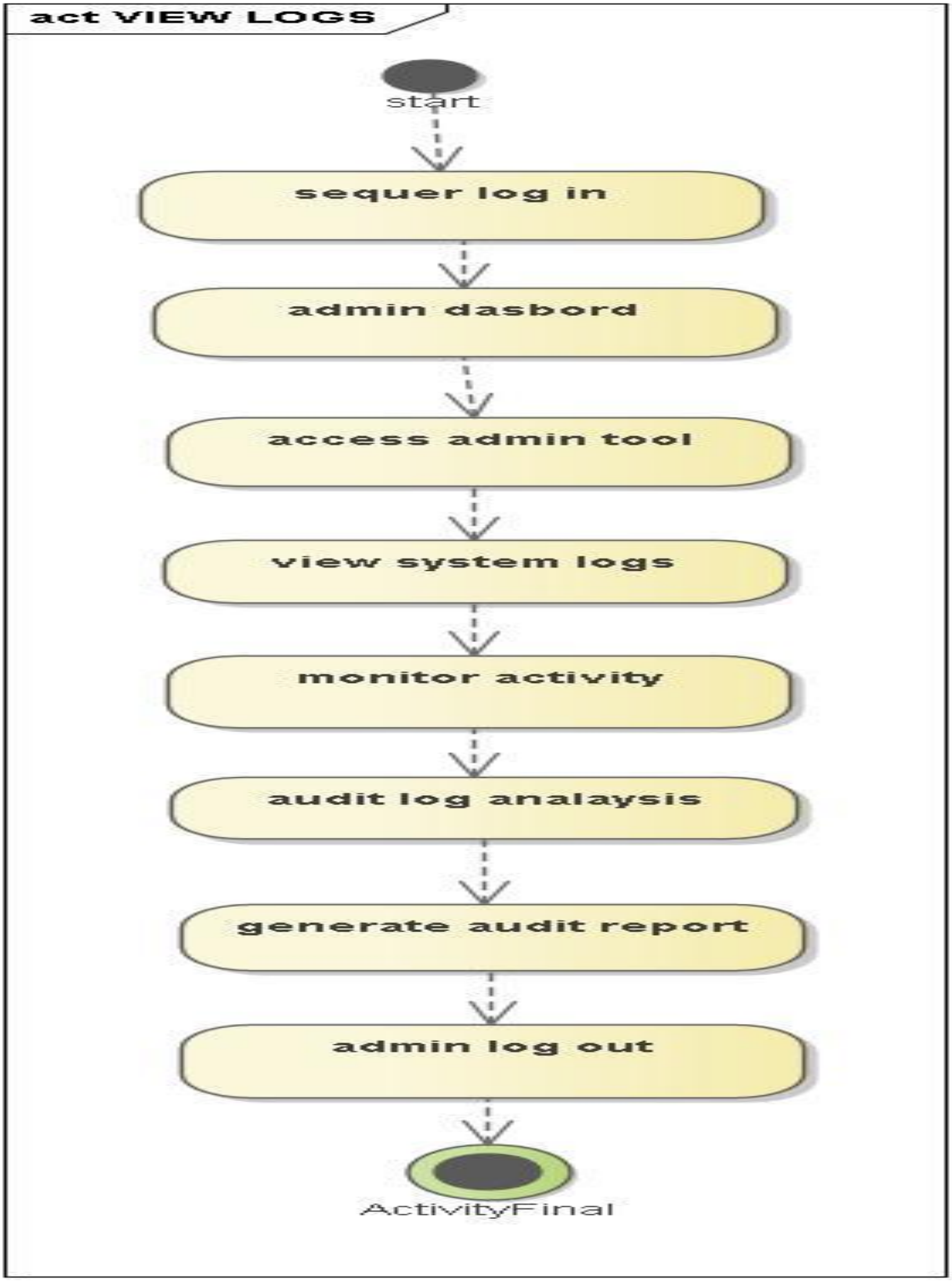


FIGURE2.21 ACTIVITY DIAGRAM VIEW LOGS

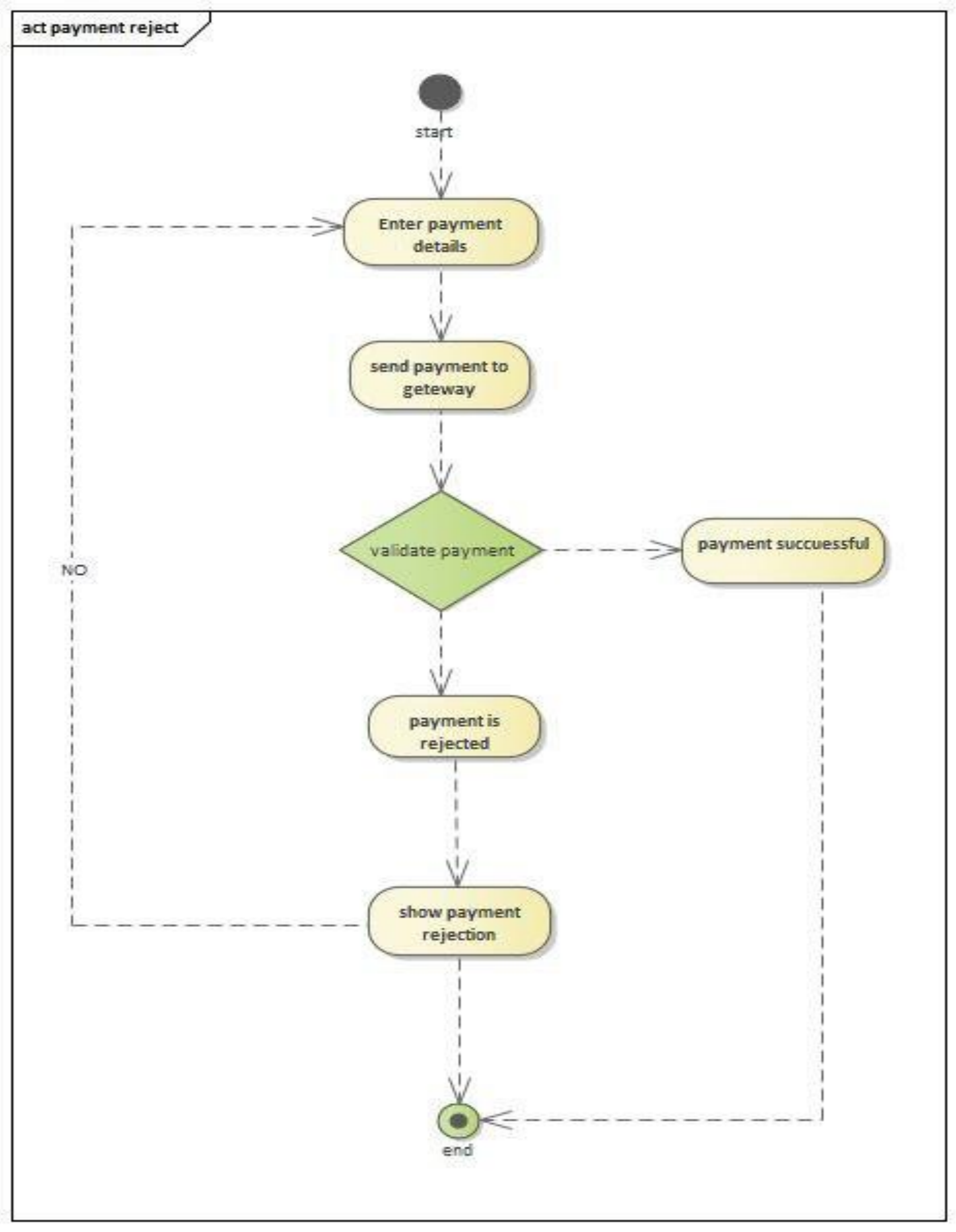


FIGURE2.22 ACTIVITY DIAGRAM PAYMENT REJECTION

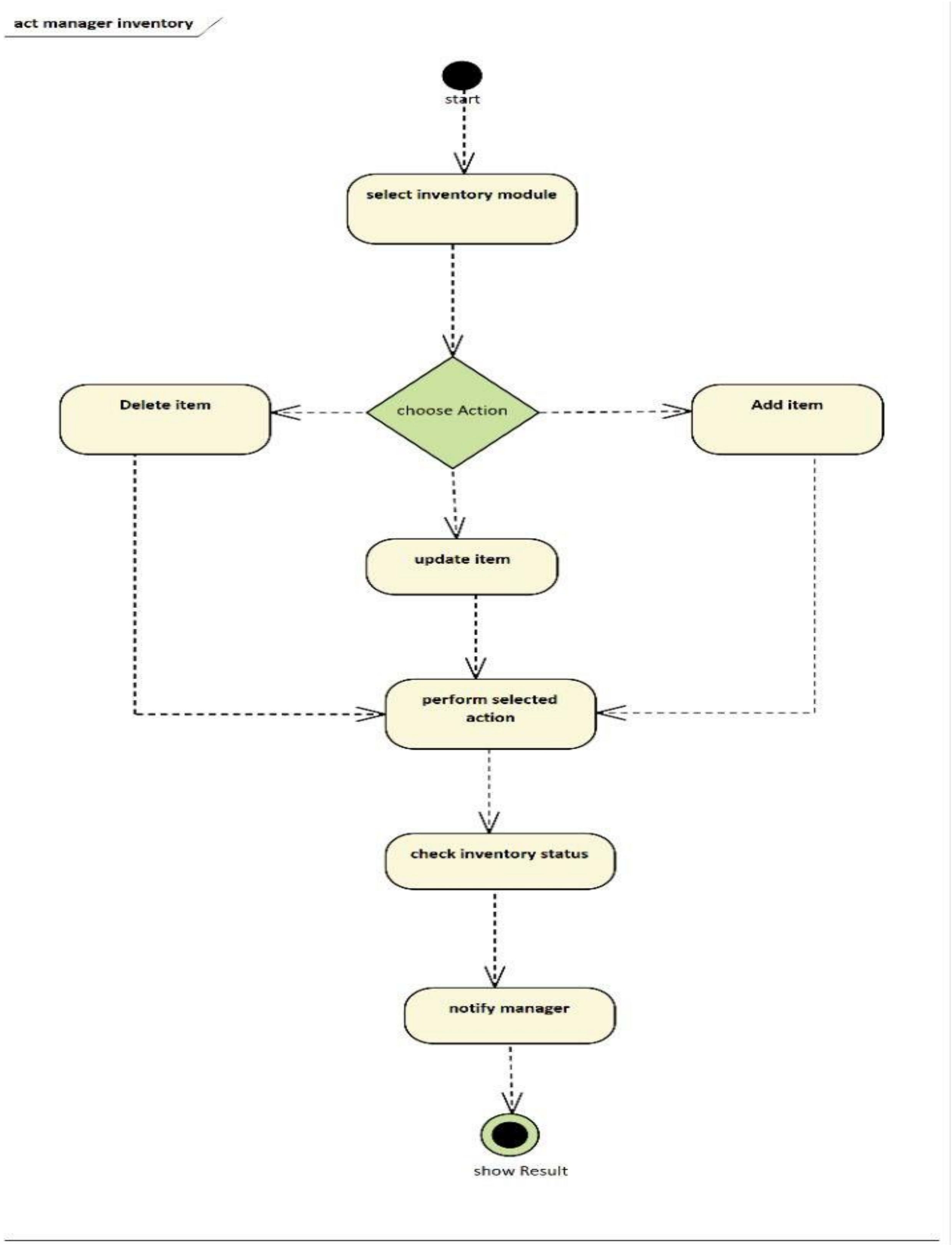


FIGURE2.23 ACTIVITY DIAGRAM MANAGE INVENTORY

2.6.6.4 STATE CHART DIAGRAM

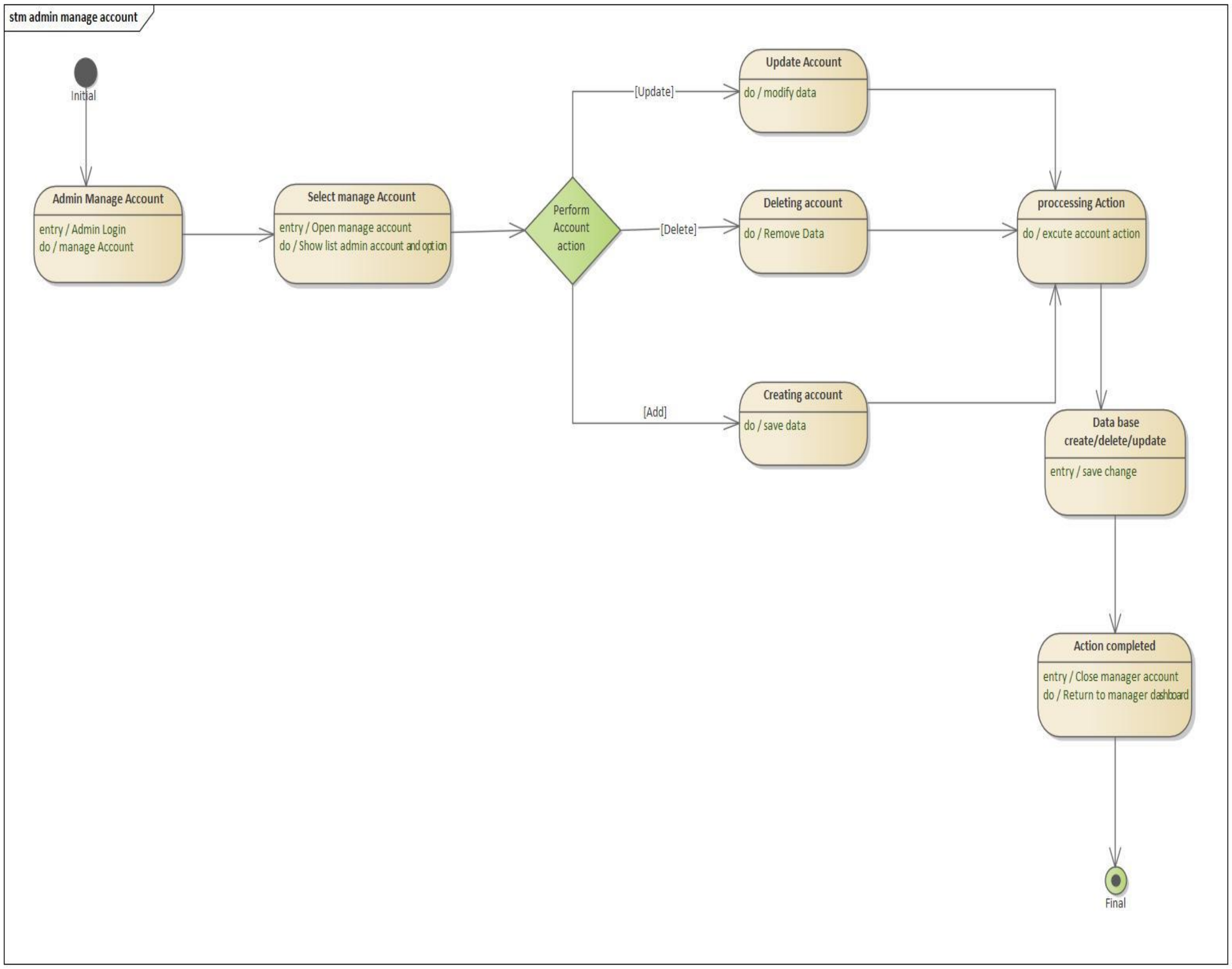


FIGURE2.24 STATE CHART DIAGRAM MANAGE ACCOUNT

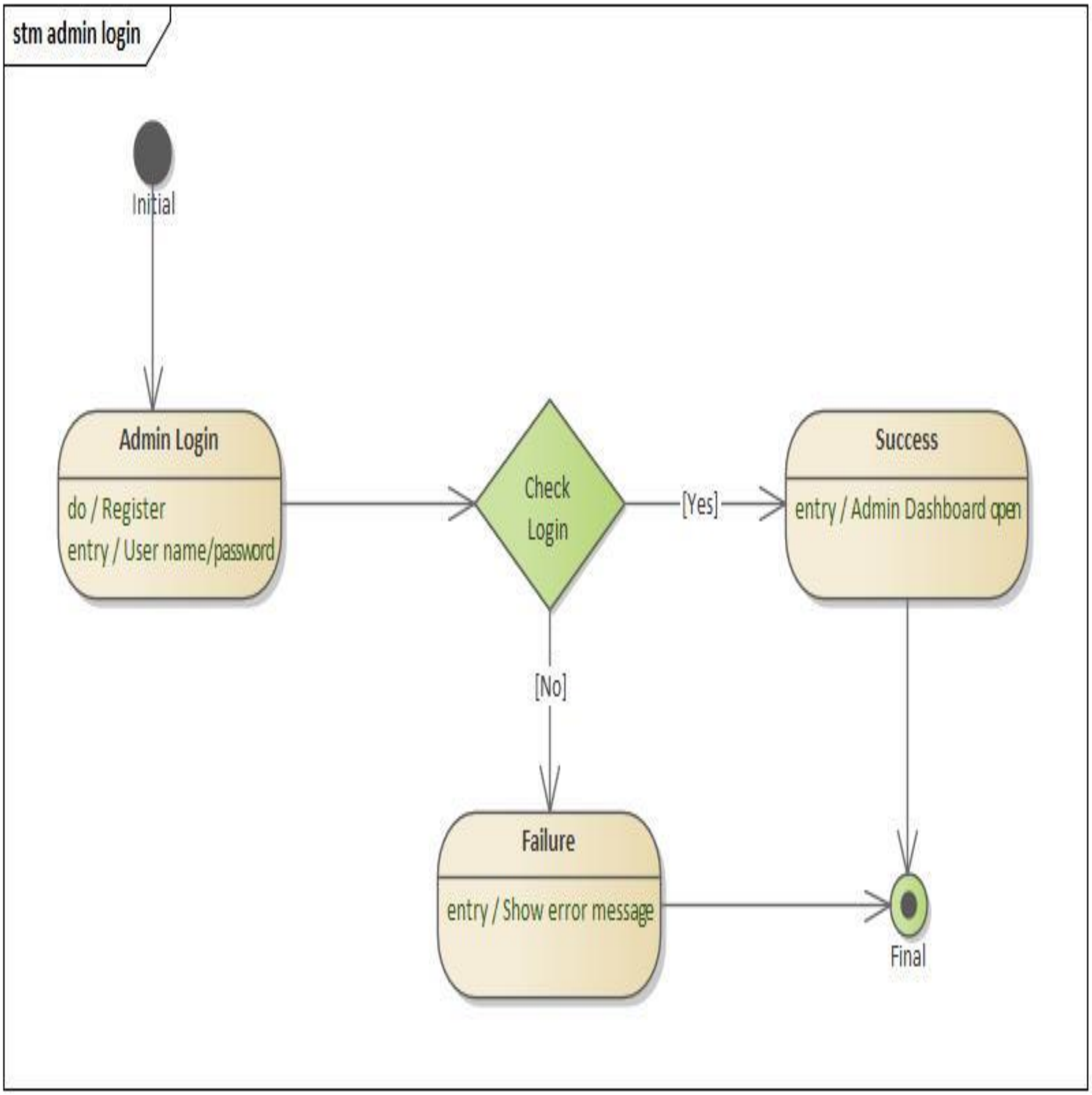


FIGURE2.25 STATE CHART DIAGRAM ADMIN LOG IN

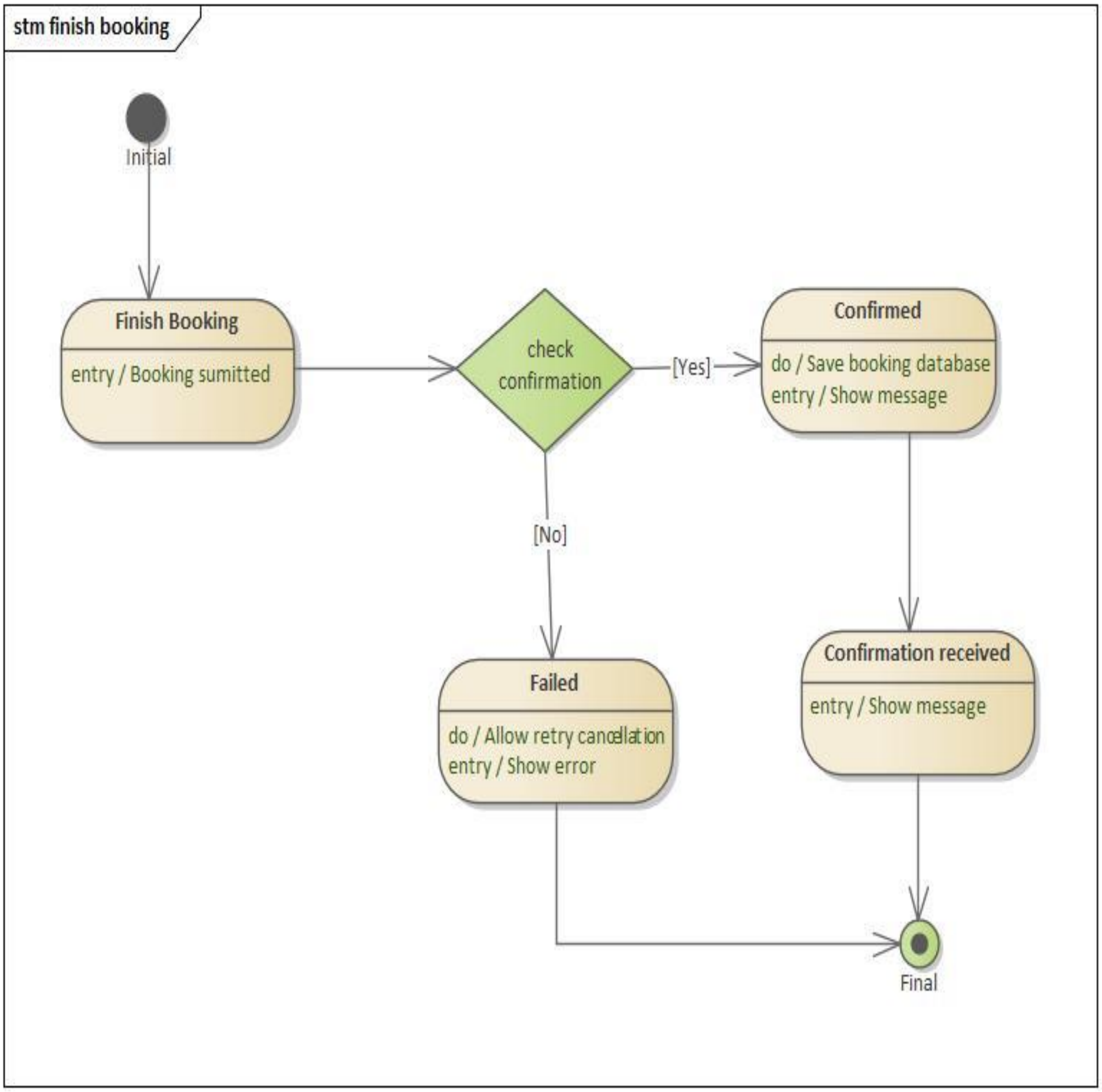


FIGURE2.26 STATE CHART DIAGRAM FINISH BOOKING

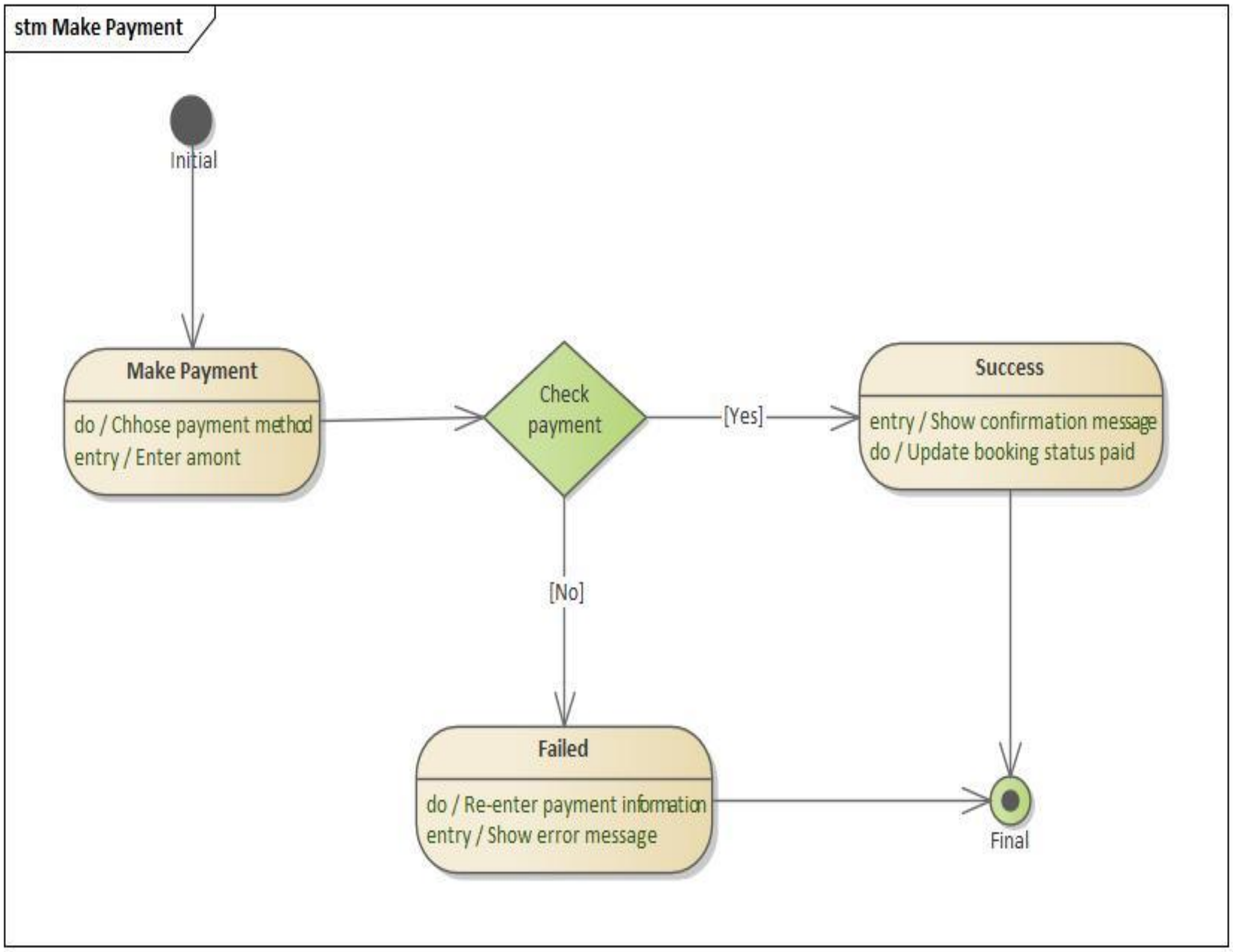

stm Make Payment
Initial
Make Payment
do / Chhose payment method
entry / Enter amont
Check payment
[Yes]
Success
entry / Show confirmation message
do / Update booking status paid
[No]
Failed
do / Re-enter payment information
entry / Show error message
Final

FIGURE2.27 STATE CHART DIAGRAM MAKE PAYMENT

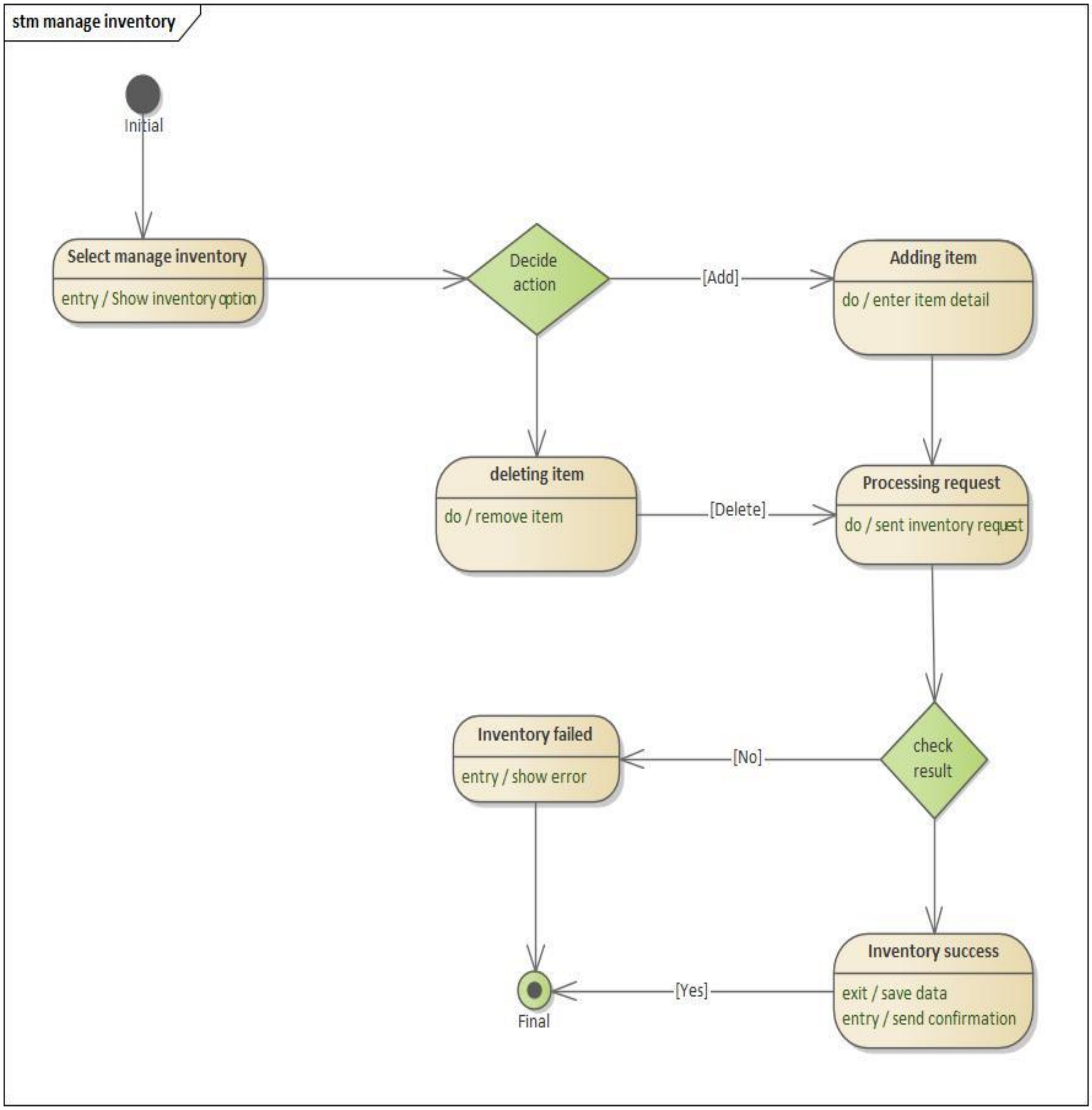


FIGURE2.28 STATE CHART DIAGRAM MANAGE INVENTORY

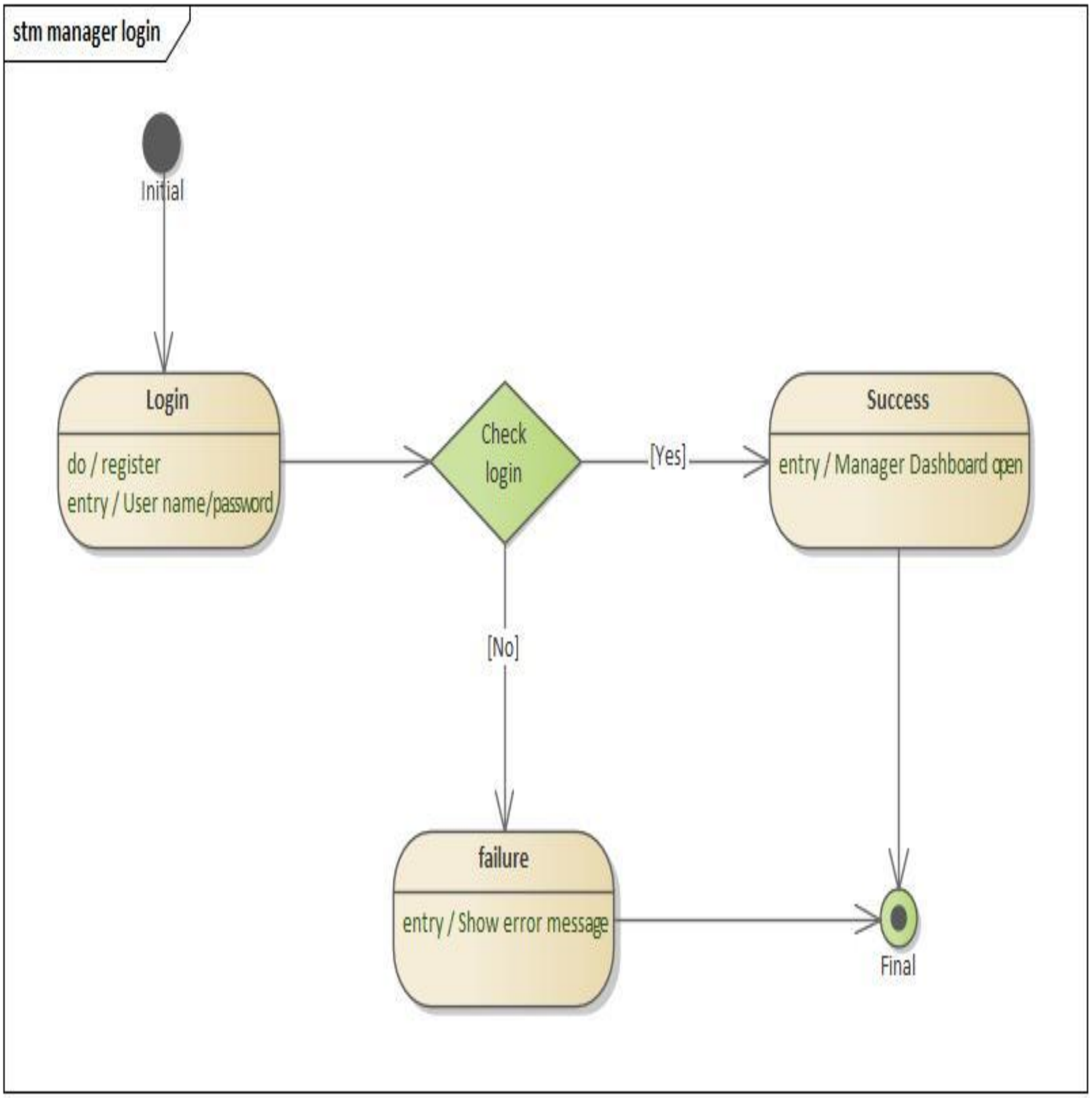

stm manager login
Initial
Login
do / register
entry / User name/password
Check login
[Yes]
Success
entry / Manager Dashboard open
[No]
failure
entry / Show error message
Final

FIGURE2.29 STATE CHART DIAGRAM MANAGER LOG IN

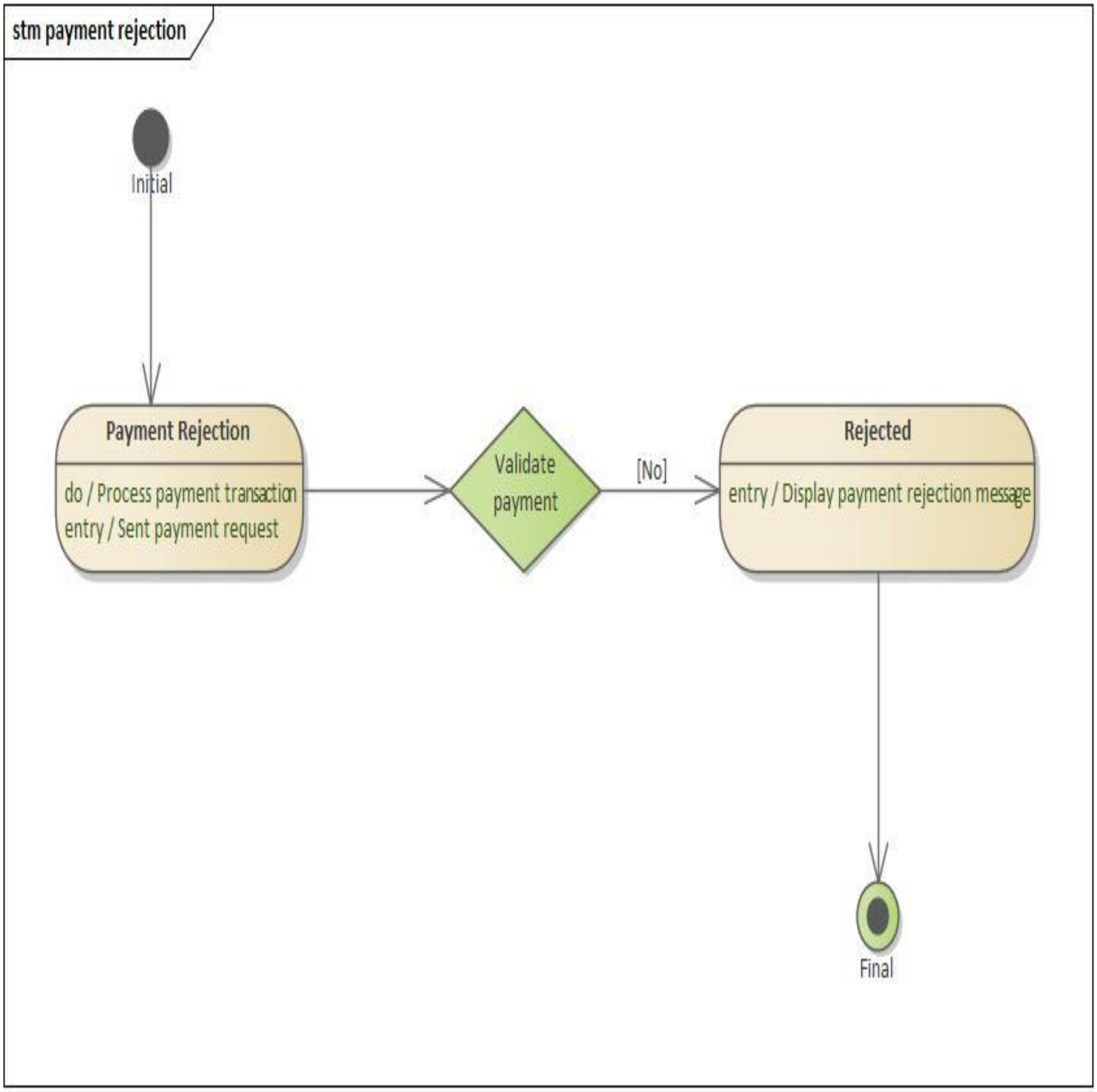

stm payment rejection
Initial
Payment Rejection
do / Process payment transaction
entry / Sent payment request
Validate payment
[No]
Rejected
entry / Display payment rejection message
Final

FIGURE2.30 STATE CHART DIAGRAM PAYMENT REJECTION

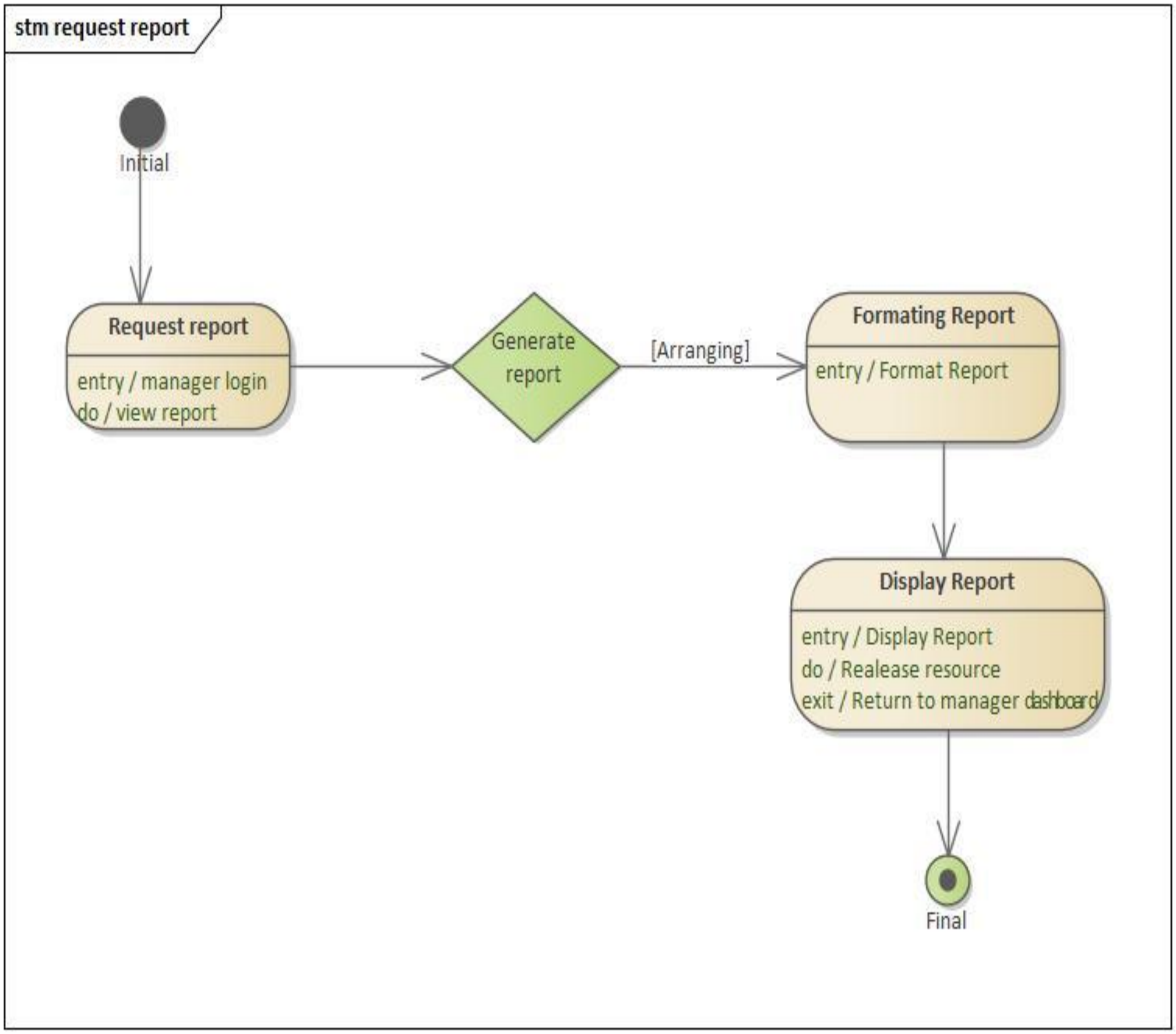


FIGURE2.31 STATE CHART DIAGRAM REQUEST REPORT

=

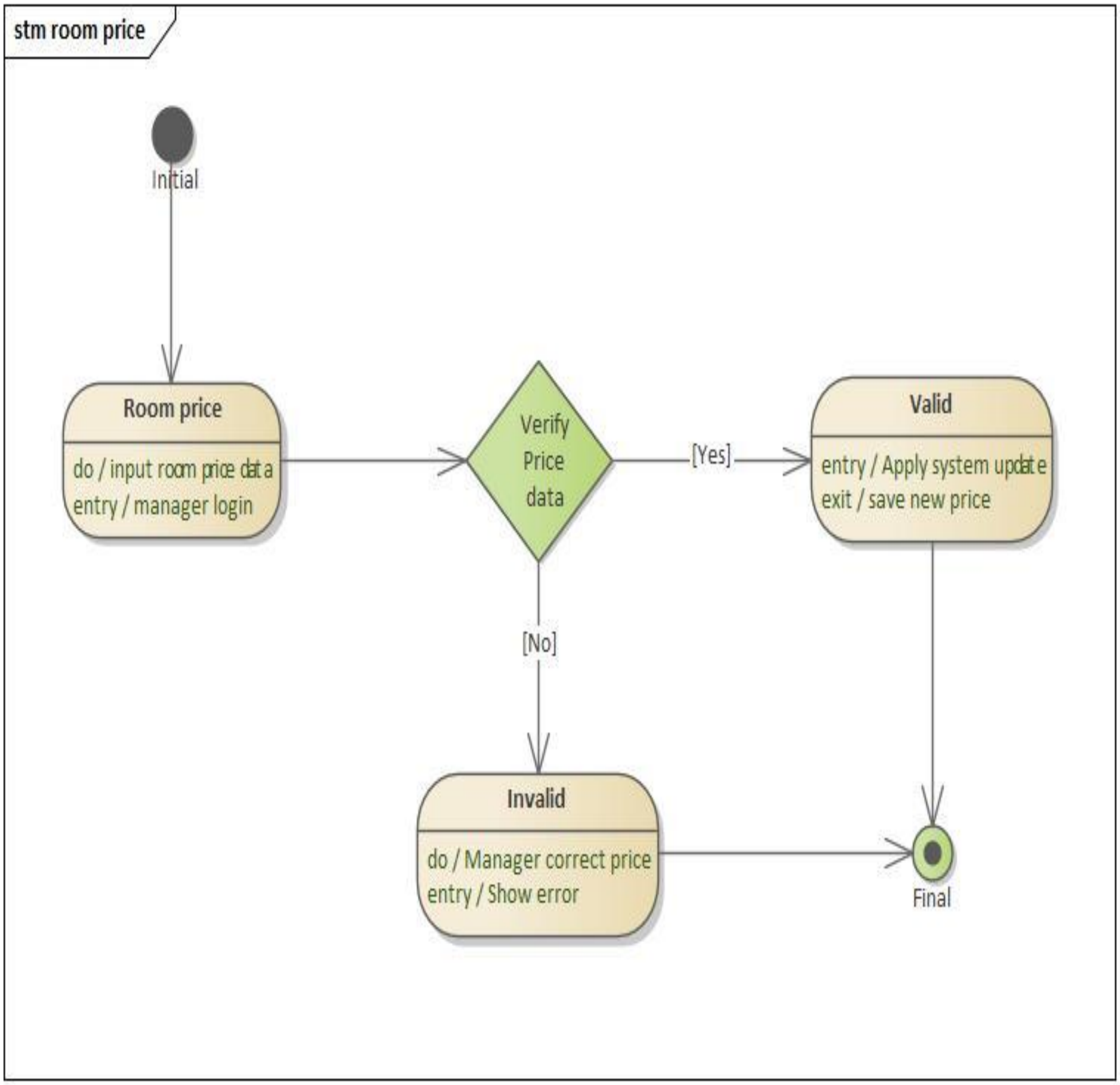


FIGURE2.32 STATE CHART DIAGRAM REQUEST REPORT PRICING

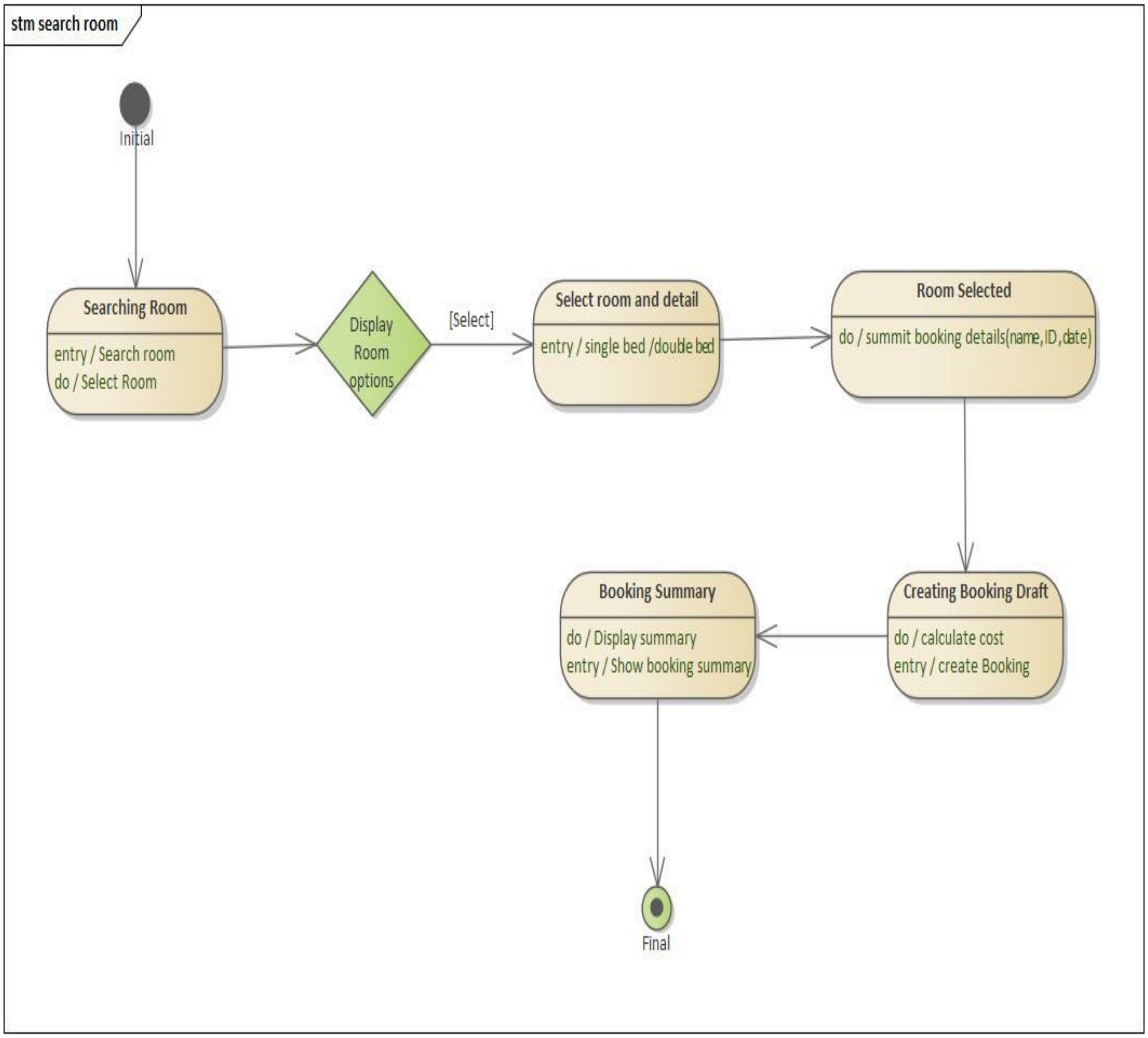


FIGURE2.33 STATE CHART DIAGRAM REQUEST REPORT SEARCH ROOM

2.6.6.5 CLASS DIAGRAM

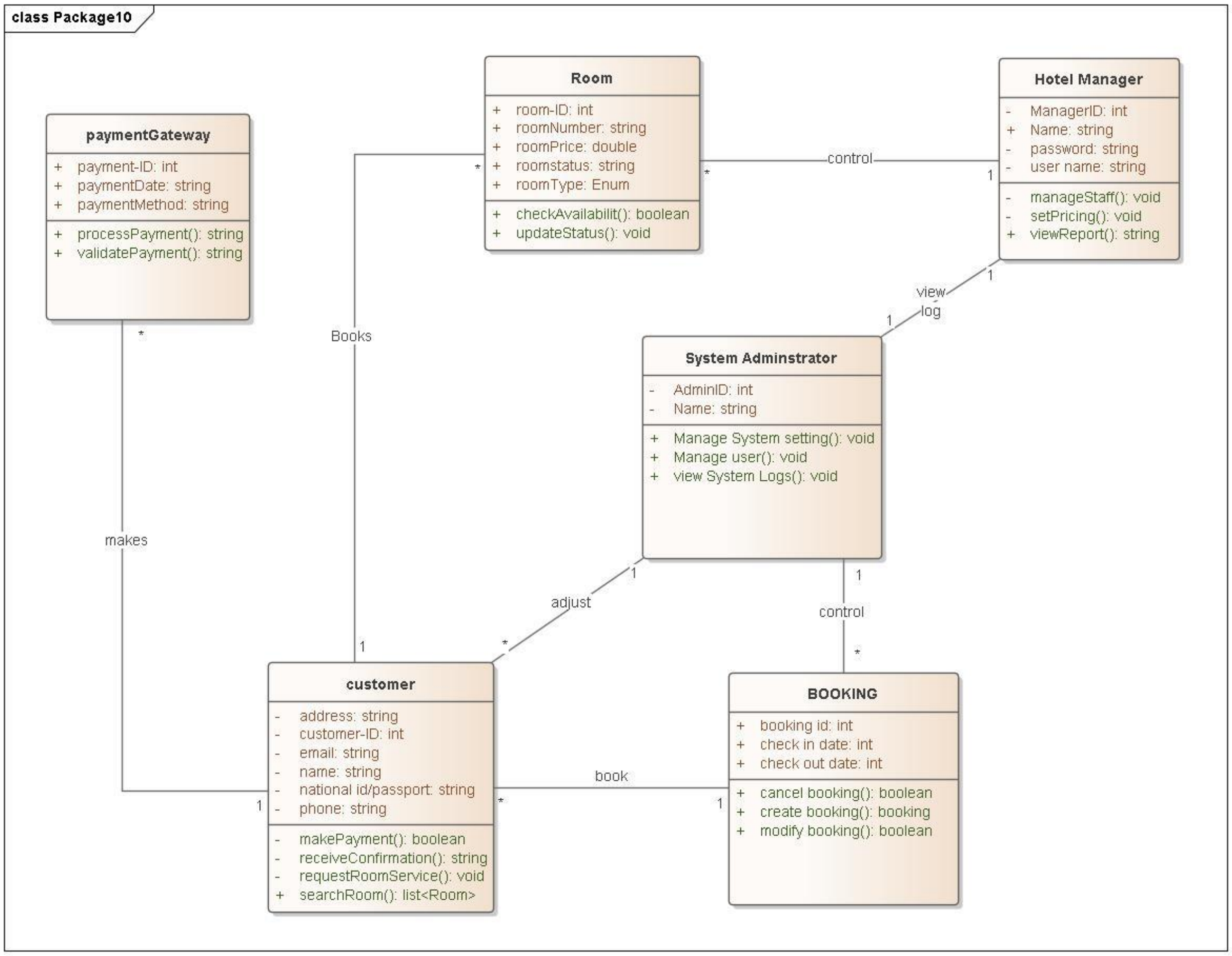
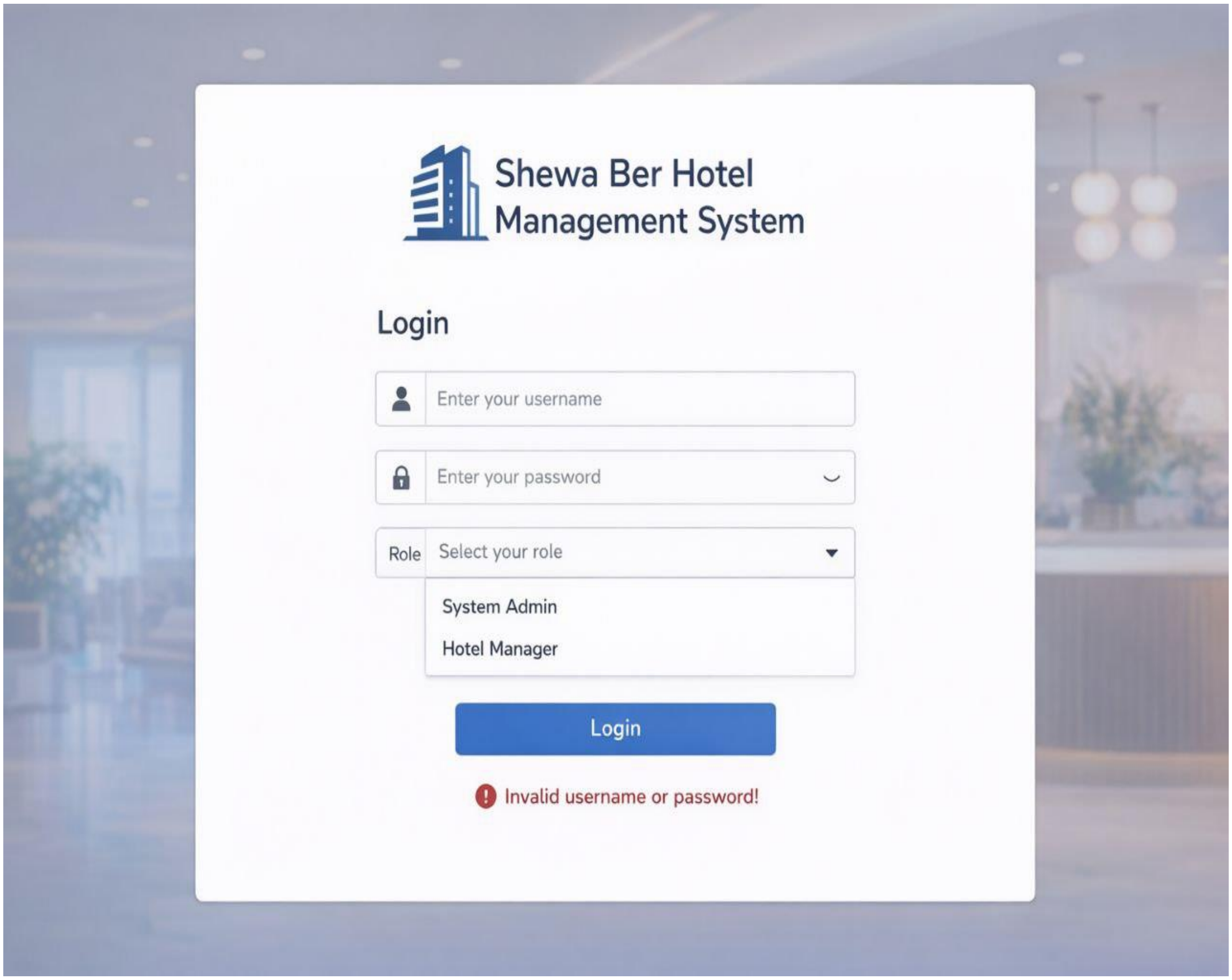


FIGURE2.34 CLASS DIAGRAM DIAGRAM

2.6.6.6 USER INTER FACE PROTOTYPING



Invalid username or password!

FIGURE 2.35 USER INTER FACE PROTOTYPE

CHAPTER THREE : DATABASE DESIGN

3.1. CONCEPTUAL DATABASE DESIGN OF THE NEW SYSTEM

3.1.1. ENTITIES IDENTIFICATION AND DESCRIPTION

1. customer : A person who makes a booking or stays in the hotel.

2. Room : A physical room in the hotel that can be booked.

3.booking : it’s make room reservation .

4. Hotel manager : oversees operation of the hotel, manage staff, set price , view report .

5. Payment gateway : handles payments made by customers.

6.System administrator : control system setting ,user account ,view system logs.

3.1.2. ATTRIBUTES IDENTIFICATION AND DESCRIPTION

Customer :

· Customer ID (PK)

· Full Name

· Address

· Phone Number

· National ID/Passport Number

· Email

Room :

· Room Number

· Room id(pk)

. manager id

· Type

· Price Per Night

Booking :

· Booking ID (PK)

· customer ID (FK)

· Check In Date

· Check Out Date

Payment gateway :

. payment id (pk)

.payment date

. payment method

.customer id

Hotel manager:

. Manager id (pk)

.Admin id (FK)

. Name

. User name

. password

System administrator:

. Admin ID(pk)

. Name

. Username

. password

3.1.3. RELATIONSHIP IDENTIFICATION AND DESCRIPTION

1. Customer makes Booking (1:N)

· One customer can make many bookings over time.

· One booking belongs to one customer.

2.Customer makes payment (1:N)

.one customer can make many payment

.one payment belongs to one customer.

3.Customers can access one room (1:1). one customer can access money rooms.

4.Manager manage room(1:N)

.one manager can manage many rooms

.many rooms can manage by one manager

5.System admin view log manager (1:1)

.one system admin control one manager system log.

6.system admin adjust customer(1:N)

.one system admin can configure/fix many customer system issue

7.System admin control booking(1:N)

.one system admin can control many booking.

3.1.4. ER DIAGRAM

.The ER diagram below summarizes the entities, attributes, and relationships described above:

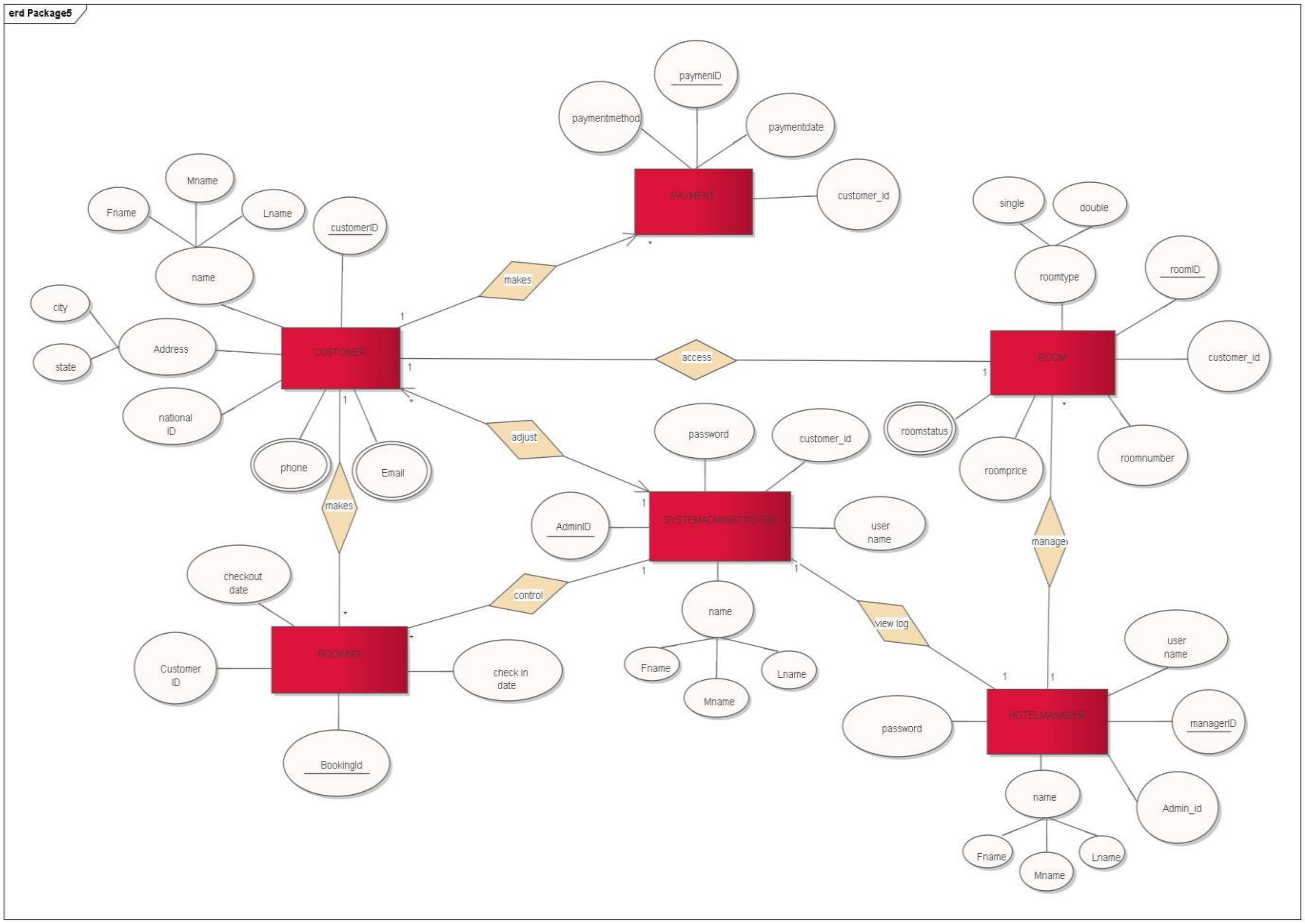
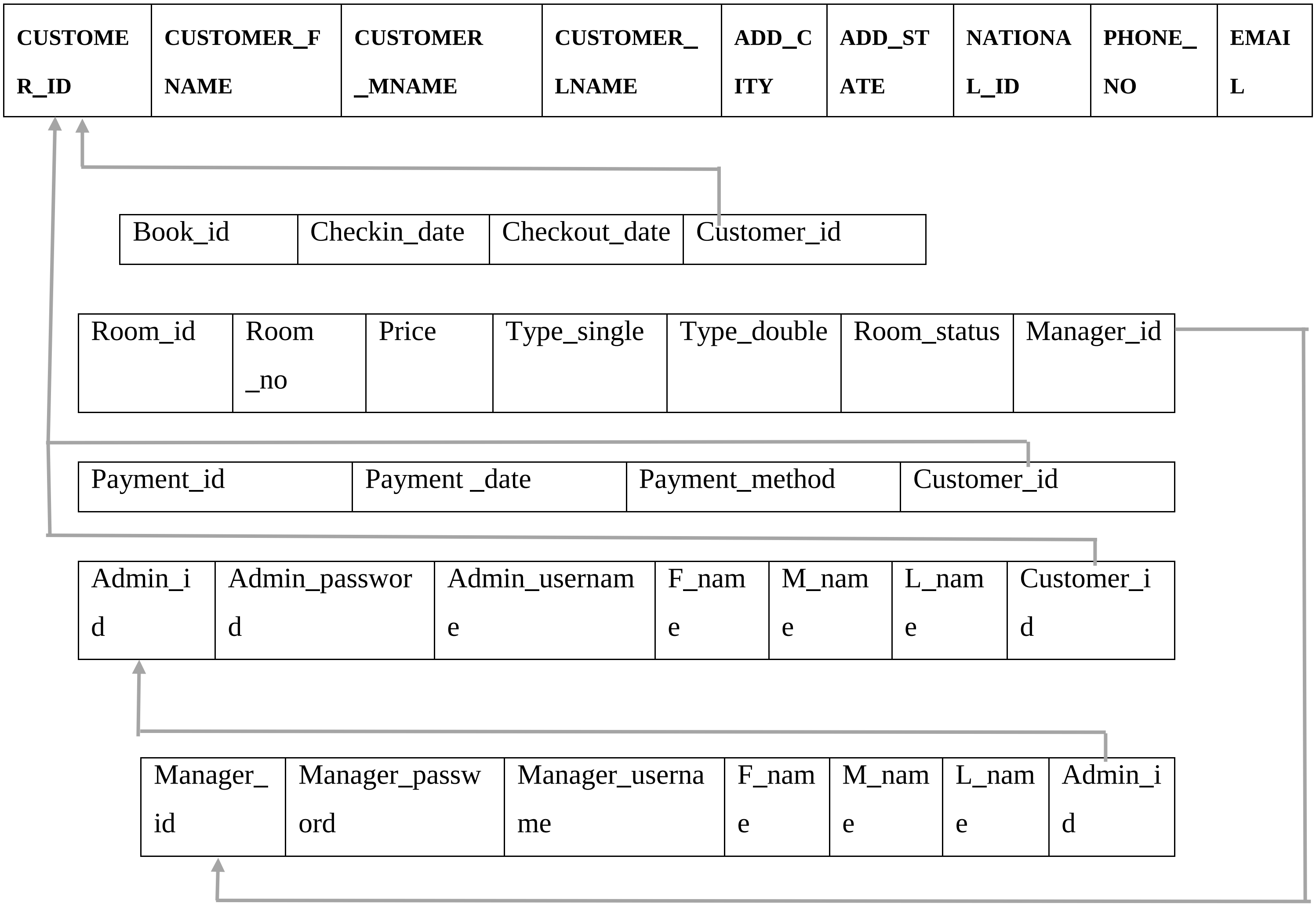


FIGURE3.1 ER DIAGRAM

3.2 LOGICAL DATA BASE DESING OF THE NEW SYSTEM

3.2.1 ER TO TABLE MAPPING



CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_MNAME	CUSTOMER_LNAME	ADD_CITY	ADD_STATE	NATIONAL_ID	PHONE_NO	EMAIL

Book_id	Checkin_date	Checkout_date	Customer_id

Room_id	Room _no	Price	Type_single	Type_double	Room_status	Manager_id

Payment_id	Payment _date	Payment_method	Customer_id

Admin_id	Admin_password	Admin_username	F_name	M_name	L_name	Customer_id

Manager_id	Manager_password	Manager_username	F_name	M_name	L_name	Admin_id

TABLE 3.1 ER TABLE MAPPING

3.2.2 VALIDATE MODULE USING NORMALIZATION

- **Customer**

Custom_id	Full_name	Address	National_id	Phone_no	Email

- **ADMIN**

Admin_id	Admin_password	Admin_username	Full name	Customer_id

- **MANAGER**

MANAGER_ID	Manager_pass	Manage_username	Full_name	Admin_id

- **ROOM**

Room_id	Room_no	price	Type_single	Type_double	Room_status	Customer_id

- **Booking**

Book_id	Checkin_date	Checkout_date	Customer_id

- **Payment**

Payment_id	Payment_date	Payment_method	Customer_id

TABLE 3.2 UN NORMALIZE TABLE

3.2.2.1 FIRST NORMALIZATION (1NF)

A. Customer

Custom_id	Custom_name	Custom_mname	Custom_lname	Add_city	Add_state	National_id	Phone_no	Email
1	beimnet	belay	sara	hosanna	Centratethiopia	1001	0993426226	Beimn@gmail.com
2	alemu	kasa	Smith	Adisababa	addis	1002	0992436272	alemu@gmail.com
3	Michael	teshe	john	hawwassa	sidama	1003	0461107676	teshe@gmail.com
4	sara	giya	gedebo	hosanna	Centralethiopia	1004	0704853473	sara@gmail.com
5	david	beckham	baba	jimma	oromia	1005	+251993426	dave@gmail.com

TABLE 3.3 CUSTOMER 1NF

✓ All attribute contain single value

- ✓ There is no repetitive group
- ✓ The table has primary key(customer_id)
- ✓ There is no duplicated row

- So this table full fill 1NF

B. Admin

Admin_id	Admin_password	Admin_username	F_name	M_name	L_name	Customer_id
101	**Asha1234**	**Ashuw**	Getachew	Mekonin	zewde	**1**

TABLE 3.4 ADMIN 1NF

- ✓ All columns contain single values
- ✓ There is no repetitive group
- ✓ The table has primary key(admin_id)
- ✓ There is no duplicated row

- There is no violation of 1NF but admin table may need some important field (role permission, phone no)

C. Manager

MANAGER _ID	Manager_pass	Manager_username	F_name	M_name	L_name	Admin_id
201	Get1234	getsh	Getachew	Mekonin	zewde	101

Table 3.5 Manager 1NF

- ✓ All columns contain single values
- ✓ There is no repetitive group
- ✓ The table has primary key(manager_id)
- ✓ There is no duplicated row

- This table full fill 1NF

D. Room

- ✓ There is no duplicated row
- ✓ The table has primary key (room_id)

- But this table violate 1NF because of type attribute split in two (type_sigle,type_double) attributes so it create "reputative group".

Room_id	Room_no	price	Types	Room_status	Customer_id

301	101	800.00	Single	occupied	1
302	102	1200.00	Double	Cleaned	2
303	103	850.00	Single	Available	3
304	104	1400.00	Double	Occupied	4
305	105	1000.00	single	Available	5

TABLE 3.6 ROOM 1NF

E. Booking

Book_id	**Checkin_date**	**Checkout_date**	**Customer_id**
401	2025-01-10	2025-01-15	1
402	2025-02-01	2025-02-05	2
403	2025-02-12	2025-02-14	3
404	2025-04-20	2025-04-25	4
405	2025-05-05	2025-05-10	5

TABLE 3.7 BOOKING 1NF

- ✓ All column contain single valued
- ✓ There is no repetitive group
- ✓ The table has primary key (booking_id)
- ✓ All column contain single value attribute

- So this full fill 1NF.

F. Payment

Payment_id	**Payment_date**	**Payment_method**	**Customer_id**
501	2025-01-09	Mobile Banking	1
502	2025-01-31	Internet banking	2
503	2025-03-11	Mobile banking	3
504	2025-04-19	Internet banking	4
505	2025-05-04	Mobile banking	5

TABLE 3.8 PAYMENT 1NF

- ✓ All column contain single value attribute
- ✓ There is no reputative group
- ✓ The table has primary key(payment_id)

- ✓ All column contain single value attribute
- ✚ So it full fill 1NF

3.2.2.2 SECOND NORMALIZATION (2NF)

A. Customer

- ✓ It's already 1NF form
- ✓ There are NO partial dependencies possible with a single-column primary key. all non key attribute fully dependent in primary key(customer_id)
- ✚ It full fill 2NF

B. Admin

- ✓ It's already 1NF
- ✓ There is no partial dependencies for this table all non key attribute fully depend on primary key (admin_id)
- ✚ It full fill 2NF

C. Manager

- ✓ It's already 1NF
- ✓ There is no partial dependencies for this table all non key attribute fully depend on primary key (admin_id)
- ✚ It full fill 2NF

D. Room

- ✓ It's already 1NF
- ✓ there is no partial dependency for this table all non key attribute fully depend on primary key (room_id)
- ✚ also it full fill 2NF

E. booking

- ✓ it's already 1NF
- ✓ there is no partial dependency for this all non key attribute fully depend on primary key (booking_id)
- ✚ it also full fill 2NF

F. payment

- ✓ it's already 1NF

- ✓ all non key attribute depend on primary key(payment_id)

- ✚ it full fill 2NF
  - ❖ over all table full fill 2nd NF becose of in our table we don’t have any composite key in our attribute

3.2.2.3 THIRD NORMALIZATION (3NF)

I. customer

- ✓ it’s already 2NF
- ✓ non primary key attribute do not depend on other non primary key attributes (no transitive dependencies) .
- ✓ but in our table city determine state so it violate 3NF rule.

Address_id	Add_city	Add_state
A1	Hossana	Central ethiopia
A2	Adisababa	Adiss
A3	Hawasa	sidama
A4	jima	oromia

Custom_id	**Custom_name**	**Custom_mname**	**Custom_lname**	**National_id**	**Phone_no**	**Email**	**Address_id**
1	beimnet	belay	sara	1001	0993426226	Beimn@gmail.com	A1
2	alemu	kasa	Smith	1002	0992436272	alemu@gmail.com	A2
3	Michael	teshe	john	1003	0461107676	teshe@gmail.com	A3
4	sara	giya	gedebo	1004	0704853473	sara@gmail.com	A1
5	david	beckham	baba	1005	+251993426	dave@gmail.com	A4

TABLE 3.9 3NF FOR A CUSTOMER

- ✓ the table full fill 3rd normalization

II. Room

- ✓ It's already 2NF
- ✓ But the table can't satisfy transitive dependency rule . in our table room type determine room price so it violate transitive dependency rule.

Type_id	Type	Price
T1	Single	800.00
T2	Double	1200.00
T3	Single	850.00
T4	Double	1400.00
T5	single	1000.00

Room_id	Room_no	price	Types	Room_status	Customer_id	Type_id
301	101	800.00	Single	occupied	1	T1
302	102	1200.00	Double	Cleaned	2	T2
303	103	850.00	Single	Available	3	T3
304	104	1400.00	Double	Occupied	4	T4
305	105	1000.00	single	Available	5	T5

TABLE 3.10 3NF FOR ROOM

- ✓ The table full fill 3NF normalization.
- Other table (booking, payment, admin, manager)are all full fill 3rd normalization form.
- Also all table full fill 4th ,5th NF

3.2.3 RELATIONAL SCHEMA WITH REFERENTIAL INTEGRITY AFTER NORMALIZATION

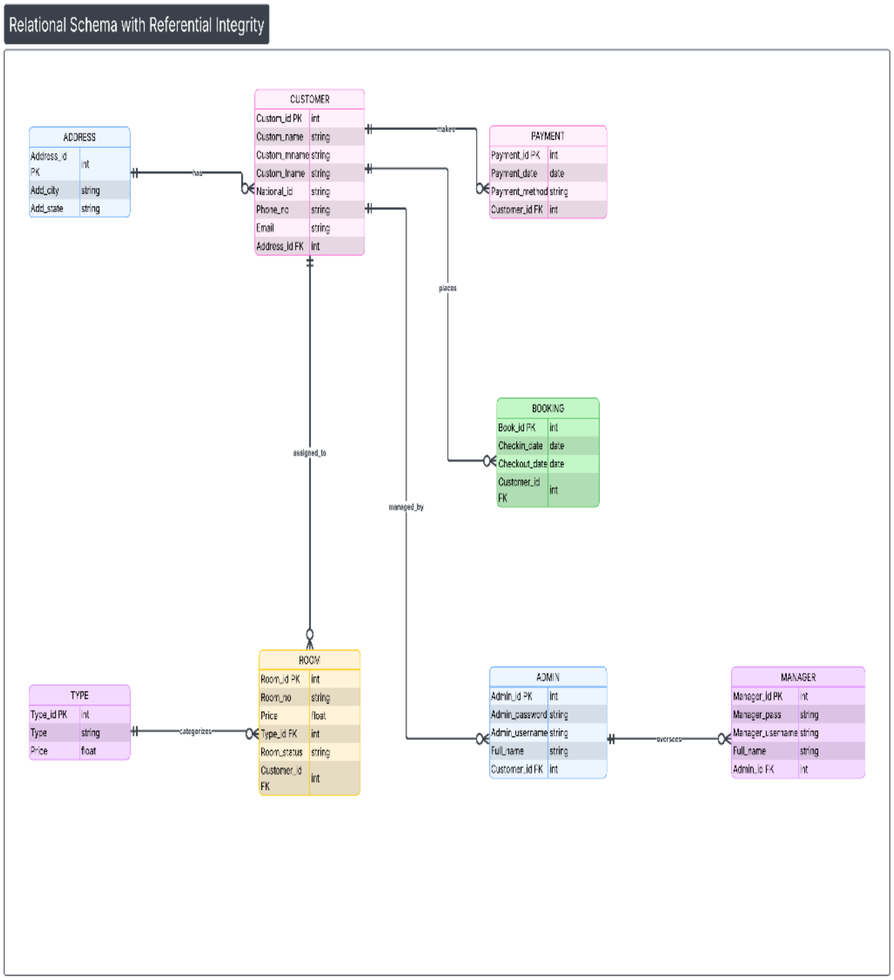


FIGURE3.2 RELATIONAL SCHEMA WITH REFERENTIAL INTEGRITY

3.3 PHYSICAL DATA BASE DESING FOR NEW SYSTEM

3.3.1 PHYSICAL DATA BASE STRATEGIES

The physical database design for the Shewaber Hotel Management System focuses on efficient storage, retrieval, and management of data while ensuring security, performance, and scalability. The following strategies

- DBMS Selection : We will use SQL as the primary relational database management system due to its reliability, scalability, and compatibility with web-based applications. SQL supports ACID transactions, ensuring data integrity and consistency
- INDEX strategy : Indexes will be created on frequently queried columns to improve search performance. primary key automatically index, Foreign Keys Indexed to speed up JOIN operations.
- Partitioning strategies : For large tables such as Booking and Payment, range partitioning will be applied based on date columns (e.g., monthly partitions) to improve query performance and simplify data archiving.
- Denormalization consideration :While the database is normalized to 3NF, controlled denormalization may be applied in reporting tables to speed up complex queries, such as daily occupancy reports.
- Backup and Recovery Strategy : Full Backups Weekly stored off site, Incremental Backups Daily to minimize data loss. Transaction Log Backups Hourly for point in time recovery.
- Security Strategy : Access Control Role-based permissions implemented at database level.

3.3.2 HARD WARE IMPLEMENTATION

Hardware implementation refers to the selection, installation, and configuration of physical devices required to run the Shewa Ber Hotel Management System efficiently and reliably. Hardware implementation refers to the process of selecting, installing, and configuring all the physical equipment required for the hotel management system to operate in the real environment. It deals with the tangible components of the system such as computers, servers, networking devices, printers, and backup power sources. These devices work together to support the software and database so that users can access the system smoothly and reliably.

Hardware implementation also focuses on establishing a reliable and secure network infrastructure. This includes connecting all computers to the internet or local network, setting up Wi-Fi access points for mobility, and applying firewalls or device-level security measures to protect the system from unauthorized access. Power management is also a key part of hardware implementation; equipment such as UPS (Uninterruptible Power Supply) is installed to protect the system from power outages and prevent data loss or hardware damage.

Overall, hardware implementation ensures that the hotel management system has a foundation of physical resources that support performance, security, and continuity. Without proper hardware implementation, even a well-designed software system may fail to perform effectively, making this stage essential for the successful deployment and operation of the hotel management system.

1. Hardware Requirements

a) Server Hardware :The server hosts the application and database.

- ✓ Processor: Intel Core i5 or higher
- ✓ RAM: Minimum 8gb
- ✓ Storage: 500 GB HDD or SSD
- ✓ Network Interface Card (NIC)
- ✓ Backup storage device

b) Client Hardware

Client computers are used by system users. Desktop or laptop computers

- ✓ Processor: Intel Core i3 or higher
- ✓ RAM: Minimum 4 GB
- ✓ Storage: 250 GB HDD or SSD
- ✓ Monitor, keyboard, and mouse.

2. Networking Hardware

- ✓ Local Area Network (LAN)
- ✓ Router and network switches
- ✓ Ethernet cables or wireless access points
- ✓ Internet connection for updates and backups

3. Peripheral Devices

- ✓ External hard drive or USB for backups
- ✓ Optional barcode or card reader

4. Power and Environmental Support

- ✓ Uninterruptible Power Supply (UPS) to prevent data loss
- ✓ Stable electrical power supply

- ✓ Proper ventilation and secure equipment placement

5. Installation and Configuration

- ✓ Install and configure all hardware devices
- ✓ Connect client systems to the server
- ✓ Test hardware functionality and network connectivity
- ✓ Ensure proper device drivers are installed

6. Maintenance and Monitoring

- ✓ Regular hardware inspection and cleaning
- ✓ Replace faulty components promptly Monitor performance to avoid hardware failure

CHAPTER FOUR : SYSTEM DESING

4.1 INRODUCTION

The system design describes how the Hotel Management System will be structured and how its components will work together to provide efficient hotel operations at Shewa Ber Hotel. The system is designed to automate daily hotel activities, reduce manual work, and improve service quality.

4.2 PURPOS OF THE SYSTEM

The purpose of the System is to automate and manage the daily operations of Shewa Ber Hotel in an efficient and reliable manner. The system is designed to reduce manual work, minimize errors, and improve the overall quality of hotel services.

The system aims to provide an easy and organized way to handle Customer information, room bookings, payments, and administrative tasks in an efficient and secure manner. It also helps hotel staff to access accurate and up-to-date information quickly.

Additionally, the system supports decision-making management by generating reports on room occupancy, revenue, and customer records.

Overall, the system is developed to improve efficiency, reliability , security, and service quality within the hotel.

4.3 DESIGN GOALS

The design goals of the system are to ensure efficiency, reliability, and ease of use while supporting the daily operations of Shewa Ber Hotel. The system is designed with the following goals.

I. Usability: Design a user-friendly system that is easy to learn and use by hotel staff such as Hotel manager, Customer, and System Admin.

II. Efficiency: Reduce manual work and processing time for room booking, payment handling, and data management.

III. Accuracy: Ensure accurate recording and retrieval of data related to customers, bookings, rooms and payments.

IV. Security: Protect sensitive data through user authentication, authorization, and role-based access control.

V. Reliability: Provide consistent system performance with minimal downtime and proper data backup and recovery mechanisms.

VI. Scalability: Allow the system to grow easily by supporting more rooms, users, and services in the future.

VII. Maintainability: Design the system in a modular way so that it can be easily updated, modified, and maintained.

VIII. Performance: Ensure fast system response time even during peak hotel operation 2 min.

4.4 CURRENT SOFTWARE ARCHITECTURE

The current software architecture of the Shewa Ber Hotel Management System is based on a manual and semi-automated approach. Most hotel operations are handled using paper records and basic computer tools, without an integrated software system.

1. Existing Architecture Description

The current system does not follow a layered or structured architecture. Instead, hotel operations are managed separately with minimal coordination. Customer information is recorded manually. Room availability is checked using paper logs or simple spreadsheets. Bookings are handled manually, which can lead to errors and double bookings. Payments are recorded using receipts and manual records. Reports are prepared manually, which is time-consuming and inaccurate.

2. System Components in the Current Architecture

User Interaction: Customers make bookings in person or by phone. Hotel managers rely on manual records to monitor operations. No centralized user interface exists.

Processing: Booking and payment decisions are made manually. No automatic validation or real-time availability checking.

Data Storage: Data is stored in paper files or basic computer documents. No centralized database. High risk of data loss and inconsistency.

3. Limitations of the Current Architecture

- High possibility of human errors
- Lack of real-time information
- Poor data security
- Difficulty in generating accurate reports
- Time-consuming operations
- No proper backup or recovery

4.5 PROPOSED SOFTWARE ARCHITECTURE

The proposed software architecture for the System of Shewa Ber Hotel is designed to improve performance, scalability, security, and maintainability. The system adopts a layered, modular architecture with support for future expansion.

I. Architecture Style

The proposed system uses a three-tier layered architecture with service-oriented modules, which separates the system into independent layers and clearly defined components.

II. Presentation Layer

This layer provides a user-friendly interface for different users of the system, including Customer, Manager, System administrator.

The interface can be web-based or desktop-based and allows users to perform operations such as Booking management, payment handling, and report viewing.

III. Application (Business Logic) Layer

This layer contains the core system logic and is divided into independent service modules:

Manager Management service.

- ➢ Booking Management Service
- ➢ Room Management Service
- ➢ Payment service
- ➢ System Administration service. Each service handles specific business rules and communicates with the database layer.

IV. Database Layer

The database layer uses a relational database management system (RDBMS) to store system data securely. It manages Customer information, Room details, Booking information, Payment transactions

User and system data. The layer ensures data integrity, consistency, and secure storage.

V. Security Architecture

- ➢ Role-based access control (Manager, Stiff-arming)
- ➢ Secure login and authentication
- ➢ Data validation and authorization at the application layer
- ➢ Regular data backup and recovery mechanisms

VI. Scalability and Integration

The architecture supports future system expansion such as:

- Online booking integration
- Mobile application access
- Payment gateway integration
- Modular services allow easy addition of new features without affecting existing modules.

7. Advantages of the Proposed Architecture

- Improved system performance
- High maintainability and flexibility
- Enhanced security
- Better user experience
- Easy scalability for future needs

4.5.1 SUB SYSTEM DECOMPOSITION

Subsystem decomposition divides the System into smaller, manageable subsystems. Each subsystem is responsible for a specific set of functions, making the system easier to develop, test, and maintain.

1. User Management Subsystem

- Responsibilities:
- User registration and login
- Role management (Manager, Customer, System administrator)
- Access control and authorization
- Outputs: Authenticated user access

2. Customer Management Subsystem

Responsibilities:

- Store and manage customer personal information
- Maintain customer history and records

Output of customer profiles and customer reports

3. Booking management Subsystem

Responsibilities:

- Create, update, and cancel Booking
- Check room availability

- Assign rooms to Customer
- Outputs: Booking confirmations and schedules

4. Room Management Subsystem

Responsibilities:

- Manage room details (room number, type, price)
- Track room status (available, occupied, under maintenance)
- Outputs: Updated room availability information

5. Payment Management sub system

Responsibilities:

- Calculate room charges and services
- Process payments
- Output: Payment records

6. Reporting Subsystem

Responsibilities:

- Generate operational and financial reports
- Provide occupancy, revenue, and performance statistics
- Outputs: Management reports

7.System Administration Subsystem

Responsibilities:

- Manage system settings.
- Manage room and users.
- Perform data backup and recovery.
- Monitor system performance and security.

4.5.2 COMPONENT DIAGRAM

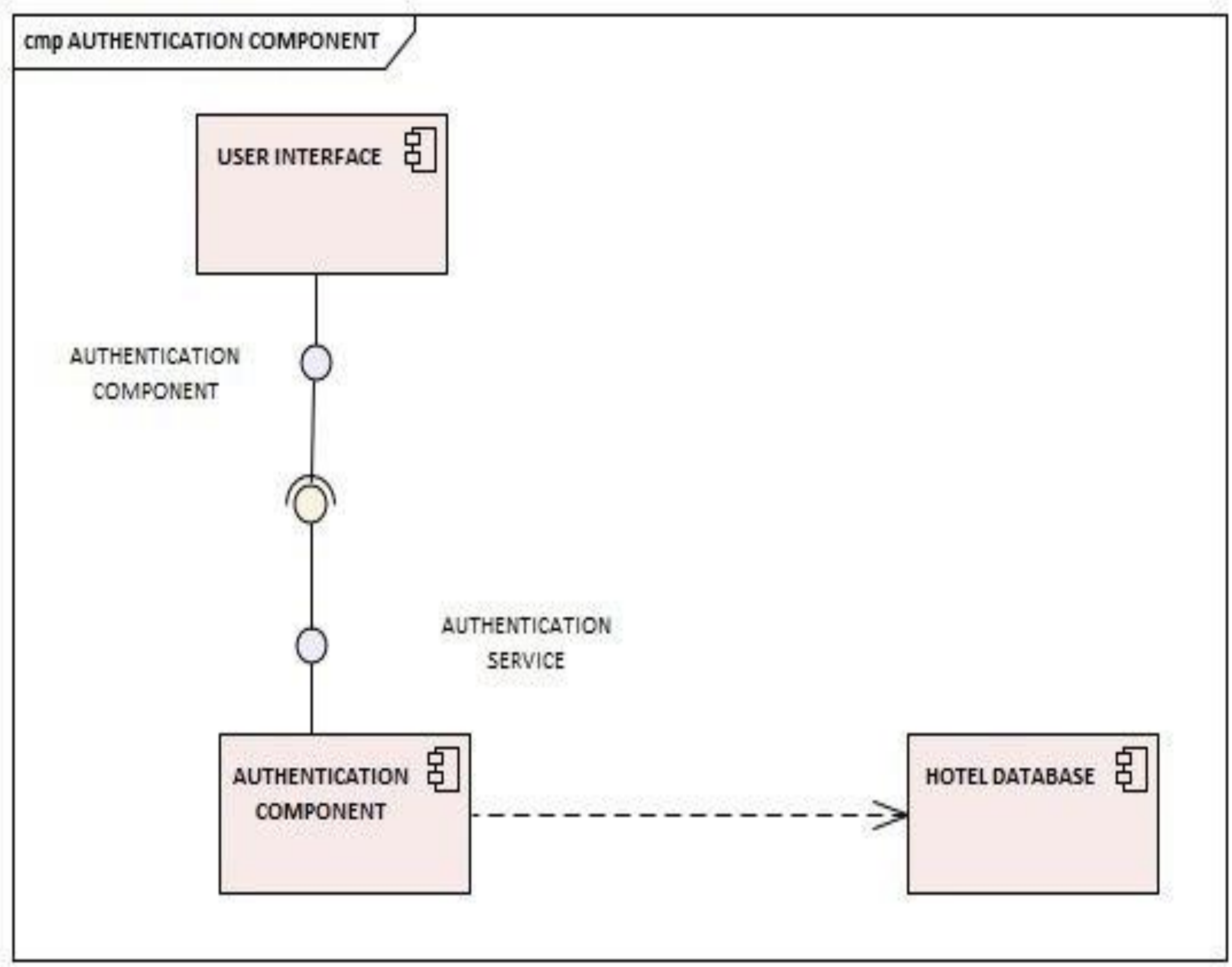


FIGURE 4.1 COMPONENT DIAGRAM AUTHENTICATION

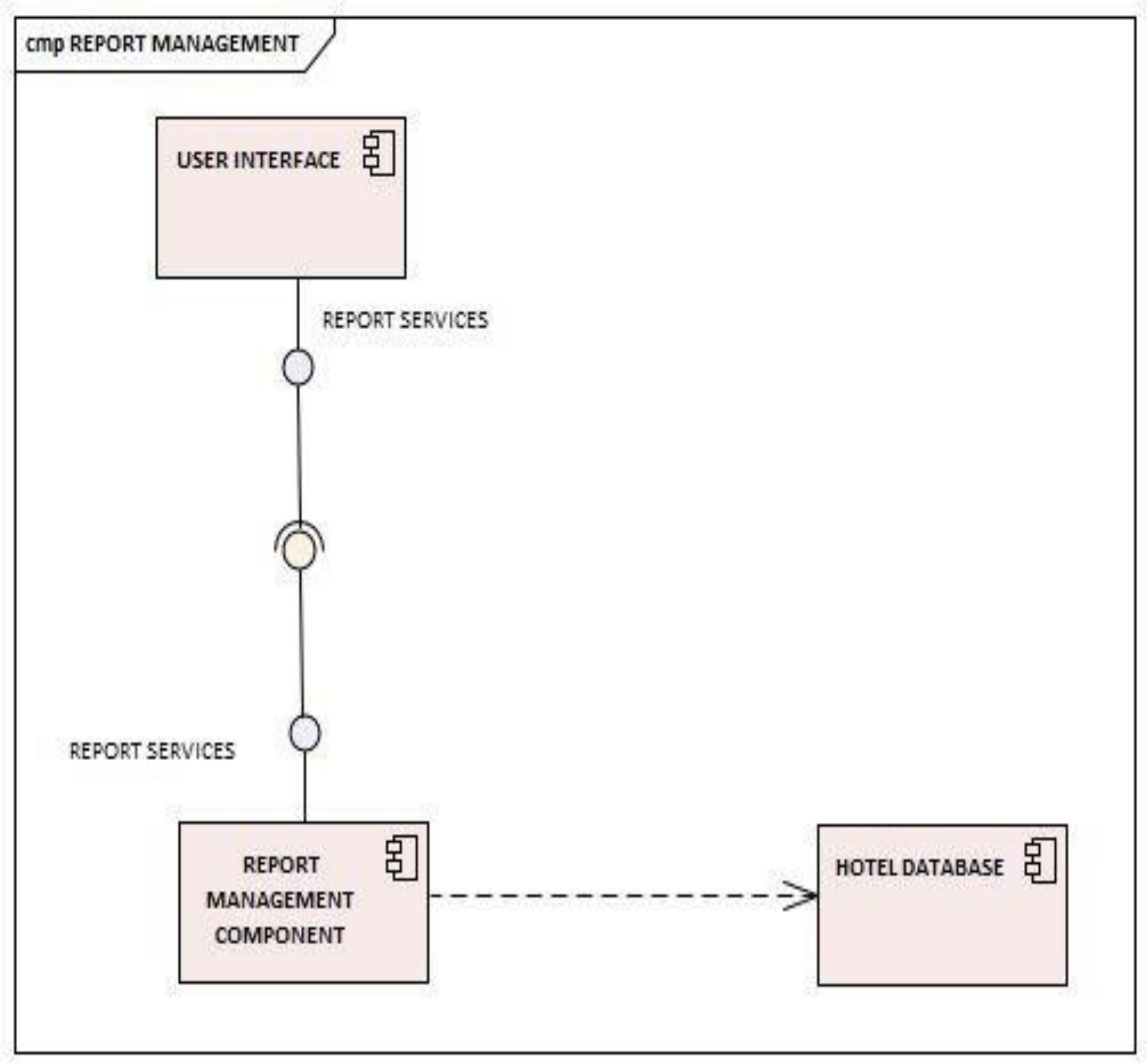


FIGURE 4.2 COMPONENT DIAGRAM REPORT MANAGEMENT

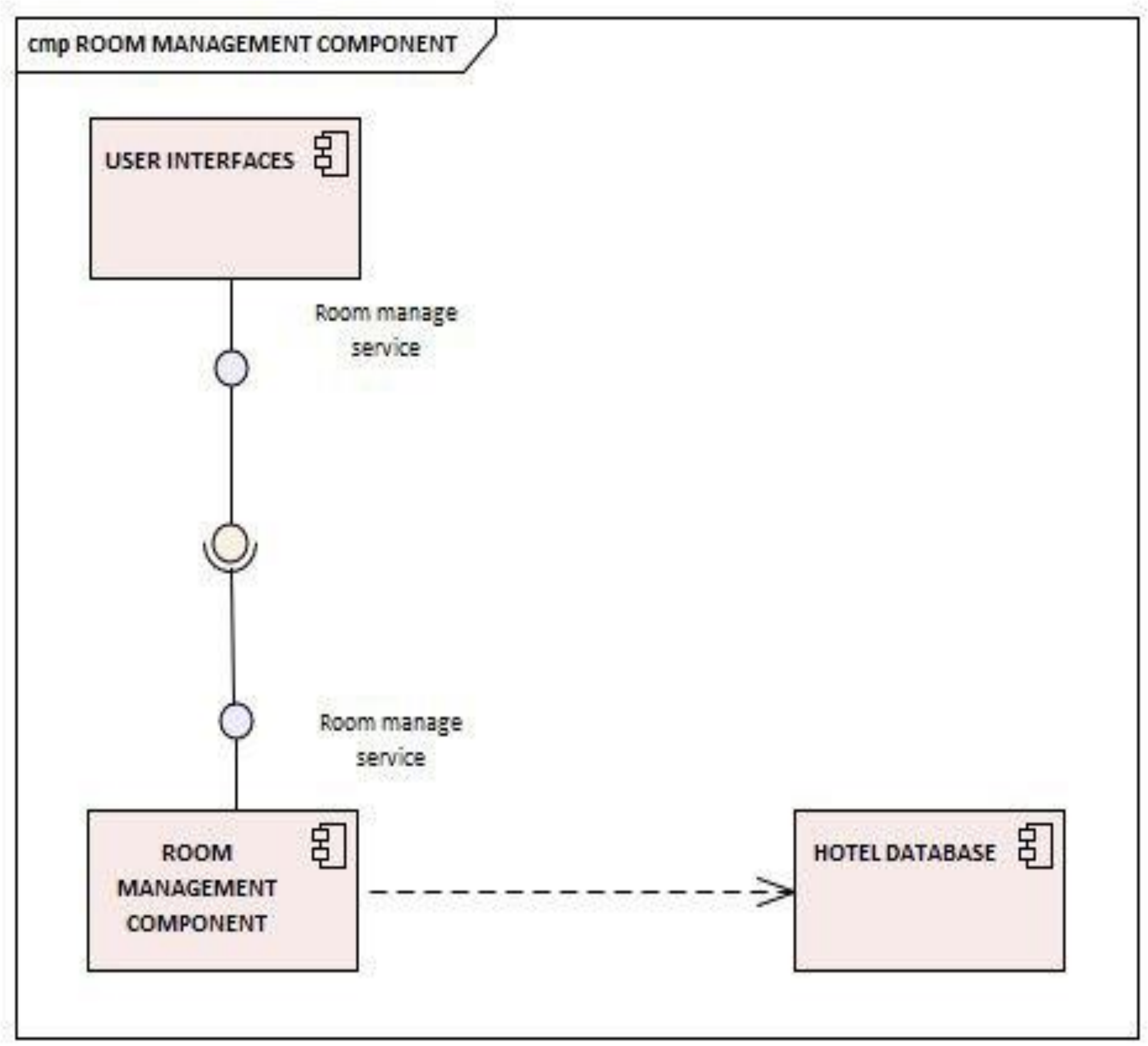


FIGURE 4.3 COMPONENT DIAGRAM ROOM MANAGEMENT

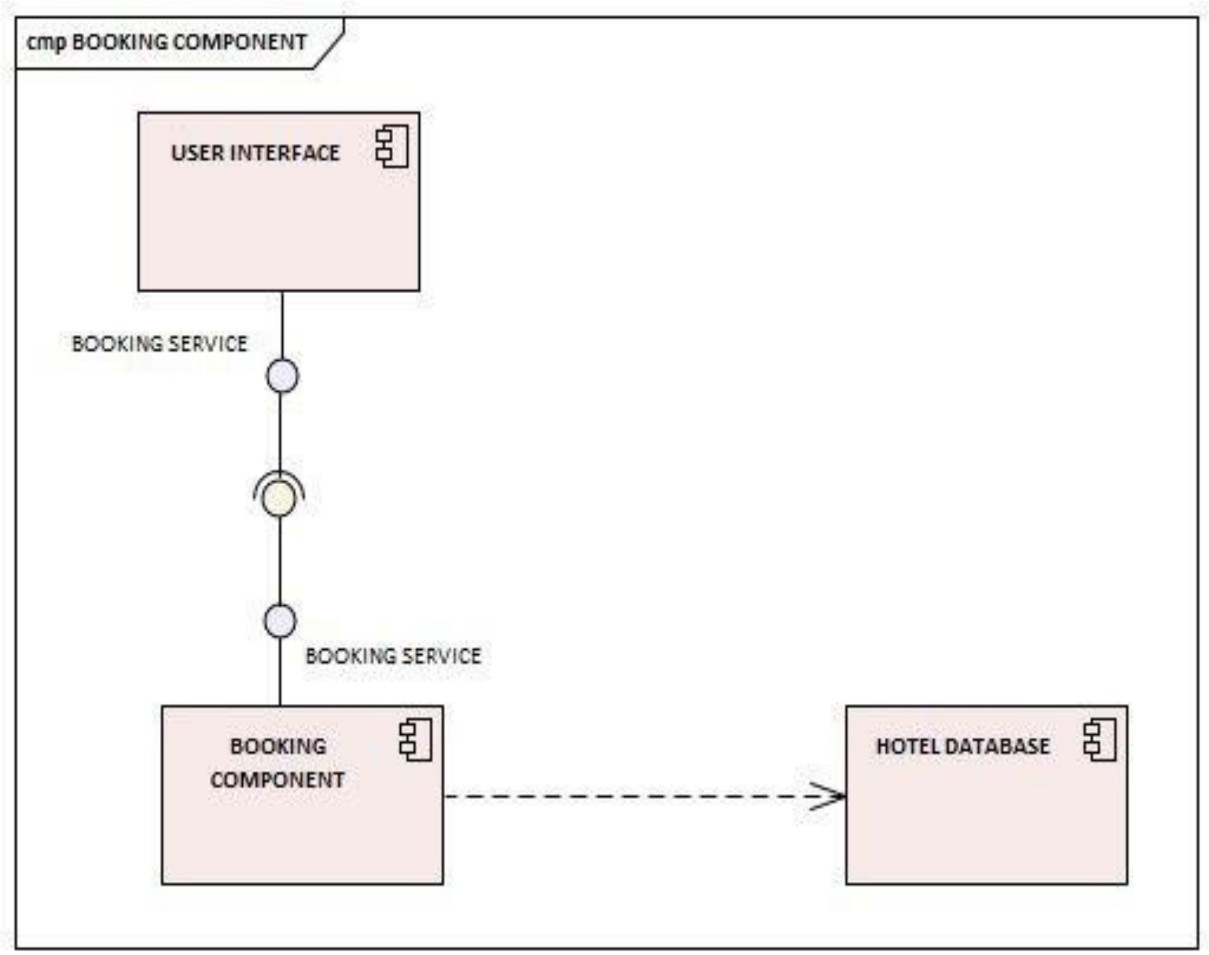


FIGURE 4.4 COMPONENT DIAGRAM BOOKING COMPONENT

4.5.3 DEPLOYMENT DIAGRAM

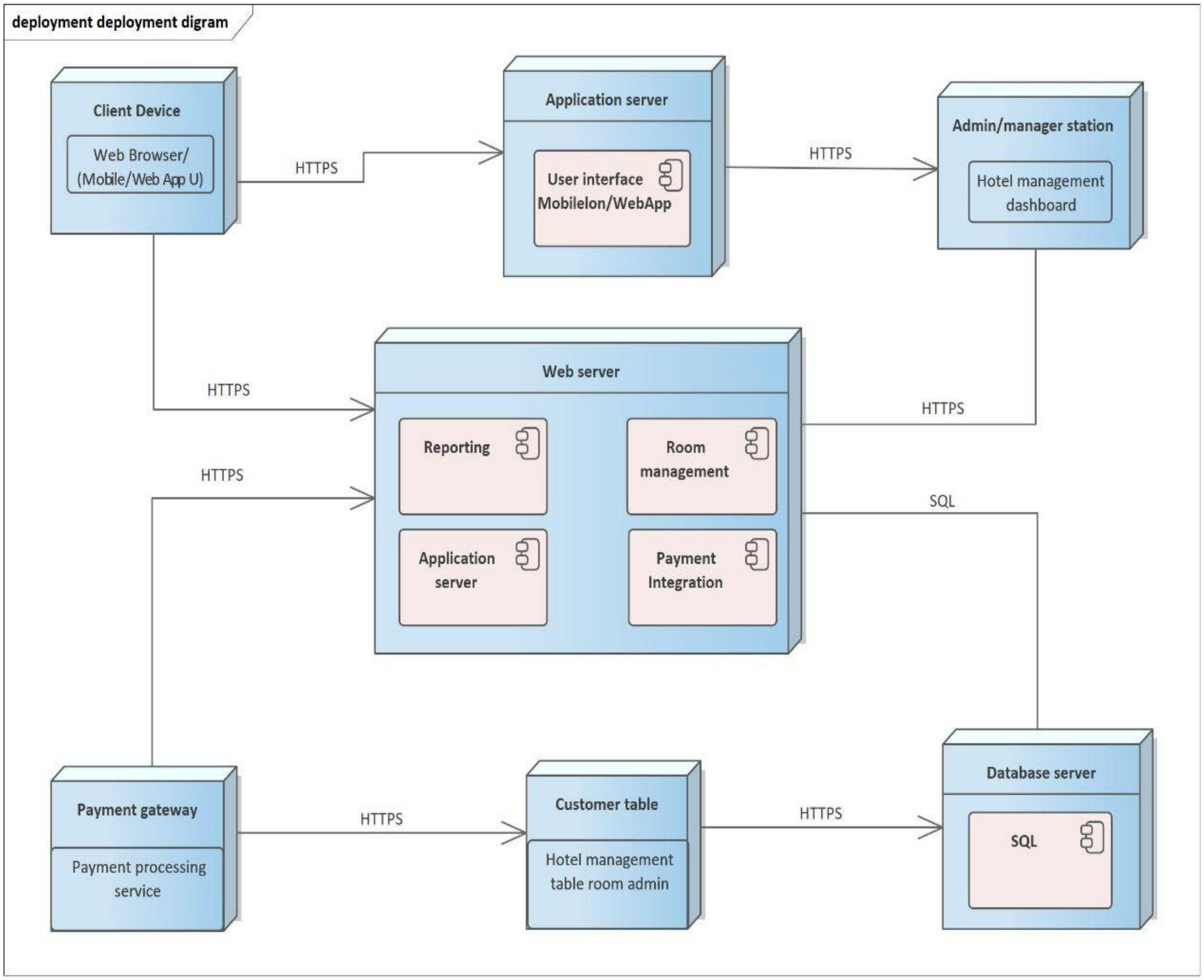


FIGURE 4.5 DEPLOYMENT DIAGRAM

4.5.4 DATABASE DIAGRAM

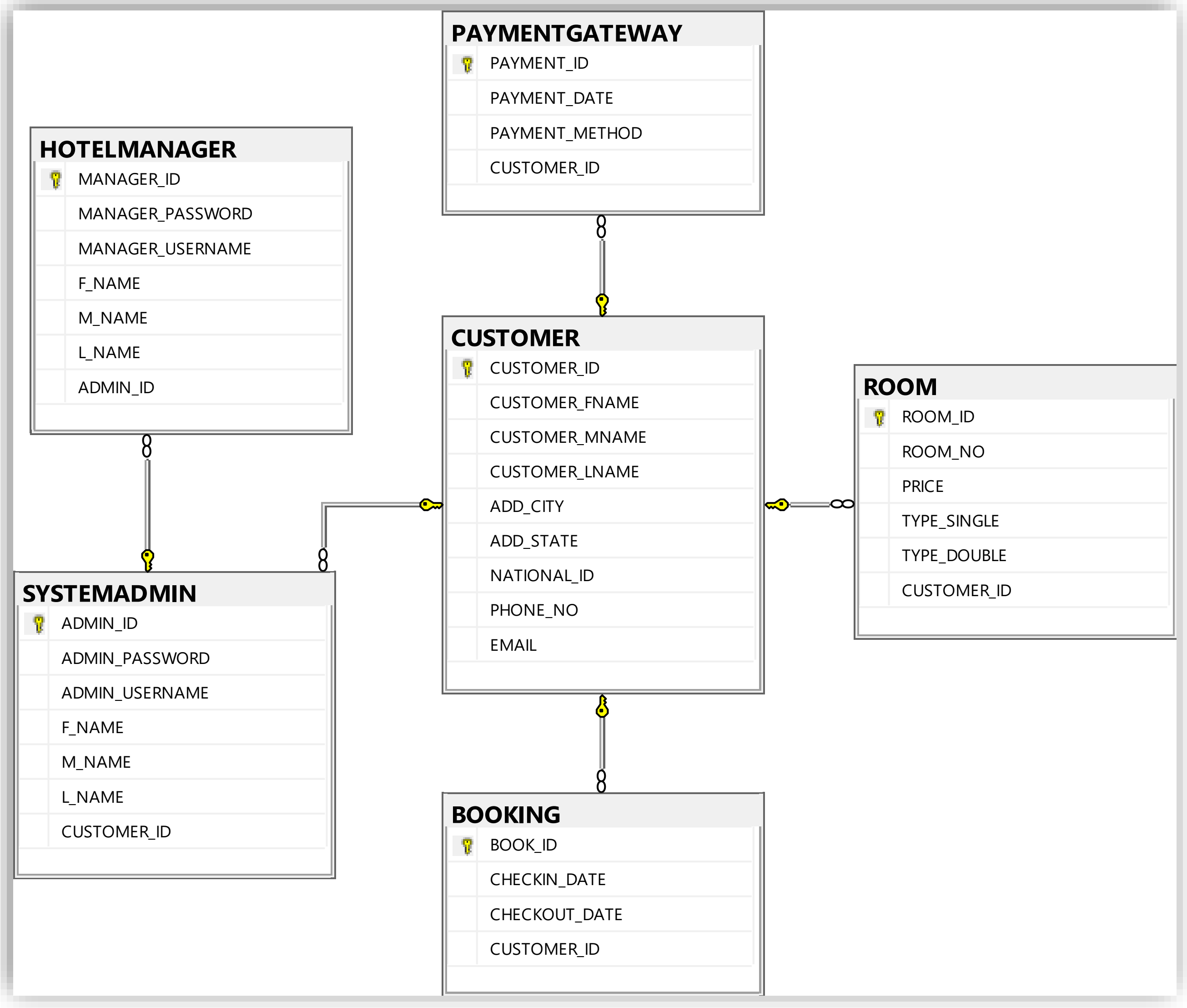


FIGURE 4.6 DATABASE DIAGRAM

4.5.5 PERSISTENT DATA MANAGMENT

Persistent Data Management refers to how System stores, retrieves, updates, and protects data so that it remains available even after the system is shut down or restarted. The system uses a relational database management system (RDBMS) to ensure reliable and long-term data storage.

1. Data Storage

All important hotel data is stored permanently in a centralized database. This includes Customer, information Room, details Booking, records Payment and transactions System users

The data is organized into well-structured tables with primary keys and foreign keys to maintain relationships between entities.

2. Data Access and Manipulation

The system accesses persistent data through the application layer using standard database operations:

- ➢ Create (insert new records)
- ➢ Read (retrieving existing data)
- ➢ Update (modify records
- ➢ Delete (remove records when necessary). This ensures controlled and consistent interaction with the database.

3. Data Integrity

- ➢ To maintain data accuracy and consistency:
- ➢ Primary and foreign key constraints are used
- ➢ Referential integrity rules are enforced
- ➢ Validation is performed before data is saved. These mechanisms prevent duplicate, invalid, or inconsistent data.

4. Transaction Management

The system uses database transactions to ensure data reliability:

- ➢ Operations are completed fully or not at all (atomicity)
- ➢ Prevents data loss during system failures
- ➢ Ensure consistency during concurrent user access

5. Security and Access Control

- ➢ User authentication and role-based access control limit data access
- ➢ Sensitive data such as passwords are stored securely

- Only authorized users can modify or delete records

6. Backup and Recovery

- Regular database backups are performed
- Recovery procedures are in place to restore data in case of system failure
- Ensures business continuity and data safety

7. Performance and Scalability

- Indexing is used to speed up data retrieval
- The database design supports future growth in data volume
- Optimized queries improve system performance

4.5.6 ACCESS CONTROL AND SECURITY

Access control and security ensure that only authorized users can access the System and that system data is protected from unauthorized use, loss, or modification.

1. User Authentication

- Every user must log in using a unique username and password.
- The system verifies user credentials before granting access.
- Failed login attempts are handled to prevent unauthorized access.

2. Role-Based Access Control (RBAC)

The system uses role-based access control to limit user actions based on their role:

Manager:

- Full access to all system modules
- View reports and manage users. This ensures users can access only the functions required for their job.

3. Authorization

- After login, the system checks user permissions before allowing actions. Unauthorized actions are blocked automatically.
- Sensitive operations (such as deleting records) are restricted to authorized roles.

4. Data Security

- Sensitive data (such as passwords) is stored in encrypted or hashed form.
- Input validation is applied to prevent invalid or malicious data entry.
- Secure database connections are used to protect data in transit.

5. Audit and Logging

- The system logs important activities such as logins, booking, and payments. Logs help track user actions and detect suspicious behavior.

6. Backup and Recovery Security

- Regular database backups protect against data loss.
- Backup files are stored securely with restricted access.
- Recovery procedures ensure system continuity after failures.

7. System Protection

- Session management prevents unauthorized session access.
- Automatic logout after inactivity reduces security risks.
- The system is protected against common threats such as unauthorized access and data manipulation.

4.5.7 GLOBAL SOFTWARE CONTROL

Global Software Control describes how the overall flow of control is managed within the System and how different components and subsystems coordinate their activities during system operation.

1.Centralized Control

The system administrator acts as the central authority for controlling system operations. All critical actions such as user management, room updates, and system configuration are handled though centralized controls.

2.Workflow Management

All transactions (booking, payment, room allocation) are controlled centrally. The system enforces sequential and dependent operations: A booking must be validated before payment.

- Payment is linked to a confirmed booking.
- Room availability updates automatically after booking completion.

3.Data Consistency Control

- The system ensures that all data changes are applied consistently across the database. Any failure during a transaction triggers rollback to maintain data integrity.

4.Security and Access Control

- Global control ensures that all users operate within their authorized roles. Access to sensitive modules (payment, user management, system configuration) is strictly monitored Logs are maintained for auditing and tracing system activity.

5.Error and Exception Handling

- The system detects errors and handles exceptions to prevent crashes or data corruption. Alerts are generated for critical errors to notify the system administrator. Errors handling is consistent across all modules, maintaining stability.

6.Monitoring and Reporting

- Global control monitors system performance and usage. Reports on bookings, payments, room occupancy, and user activity are generated for management. System monitoring helps with proactive decision making and system maintenance.

4.5.8 BOUNDARY CONDITIONS

Boundary conditions define the limits and constraints under which the system operates. They specify what the system can and cannot do and help ensure reliable operation by identifying system inputs, outputs, and interactions with external entities.

1. System Inputs

The system accepts inputs only from authorized users:

- customer: Registration details (name, contact, email)
- Booking information**:** (room selection, check-in/out dates), Payment information (amount, method)
- HOTEL MANAGER**:** Requests for reports**,** Updates to room status or bookings
- SYSTEM ADMINISTRATOR**:** User management (add/update/delete users)**,** System configuration settings, Backup and recovery operations

CONSTRAINTS: input must be valid and within predefined ranges (e.g., booking dates cannot be in the past, payment amounts must be positive).

2. System Outputs

- The system produces output only for authorized purposes: Booking confirmations and receipts, Reports for hotel management (occupancy, revenue), System logs for auditing and monitoring

CONSTRAINTS: Outputs must be consistent, accurate, and reflect the current state of the system

3. Operational Limits

- Maximum number of simultaneous bookings per room is one.
- Payment processing must be completed before confirming a booking.
- Room availability must be checked before accepting a booking.

- The system only supports the defined roles (Customer, Hotel Manager, System Administrator).

4. Interaction with External Entities

- Customers may interact via web or application interfaces.
- Payments may interact with external payment gateways (if implemented).
- External backup storage may be used for data recovery.

CONSTRAINTS All: external interactions must be secure and authenticated.

4.6 REFINED ER DIAGRAM

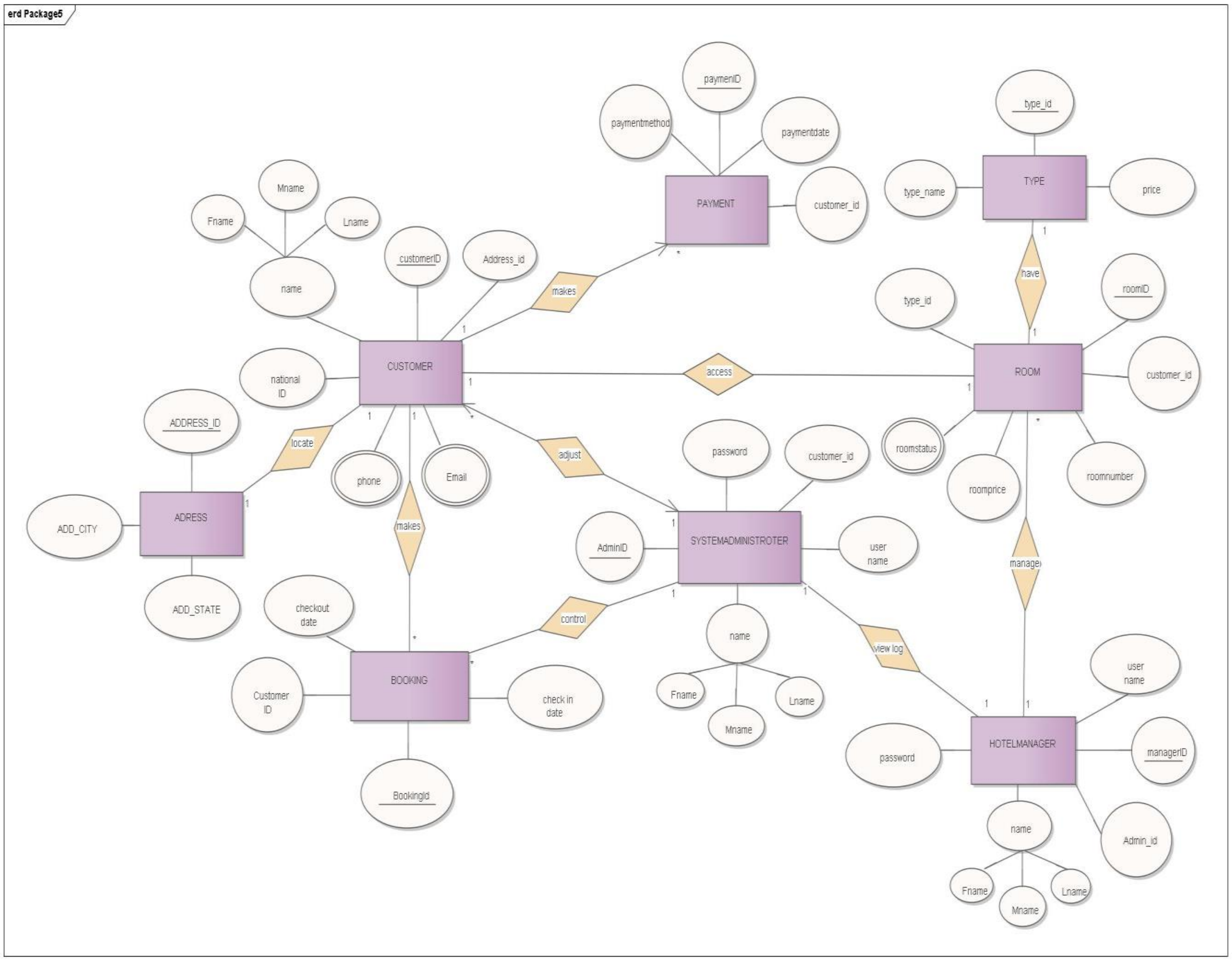


FIGURE 4.7 REFINE ER DIAGRAM

CHAPTER FIVE: IMPLIMENTATION AND TESTING

5.2 USER MANUAL PREPARATION

The user manual is prepared to guide users in effectively operating the System. It provides clear instructions, explanations, and guidelines to ensure proper system usage by all user roles.

1. Purpose of the User Manual
    - Help users understand how to use the system correctly
    - Reduce user errors during system operation
    - Provide step-by-step instructions for common tasks
2. Target Users
    - Customers – for booking rooms and making payments
    - Hotel Managers – for viewing reports and monitoring operations
    - System Administrators – for managing users, rooms, and system settings
3. Contents of the User Manual

I. System Introduction**:** Overview of the shewber hotel management system objectives and benefits.

II. System Requirement: Hardware and software requirements Supported operating systems

III. Login and Authentication: How to log in to the system, Password rules and security guidelines

IV. System Navigation**:** Explanation of menus and buttons Description of system screens

V. user Operations : booking, Payment processing ,Viewing booking history ,Security Guidelines ,Safe password practices .

4. Format of the User Manual
    - Simple and clear language
    - Step-by-step instructions
    - Screenshots and diagrams where necessary
5. Maintenance of the User Manual
    - The manual will be updated when system features change
    - Feedback from users will be used to improve clarity
    - New versions will be distributed to users after updates

5.3TRAINING AND INSALLATION

Training and installation are essential to ensure the successful deployment and effective use of the System. Proper installation guarantees correct system operation, while training ensures users can confidently and efficiently use the system.

1.SYSTEM INSTALLATION

A. Installation environment

- ✓ The system is installed on hotel computer or a central server.
- ✓ Required hardware and software are verified before installation.
- ✓ Supported operating systems and database software are prepared in advance.

B. Installation Steps

- ✓ Install the required operating system and database software.
- ✓ Deploy the System application.
- ✓ Configure system settings and database connections.
- ✓ Create initial user accounts (system administrator and hotel manager).
- ✓ Load initial data such as rooms and pricing.
- ✓ Test the system to confirm successful installation.

C. Installation Validation

- ✓ Verify system functionality through basic test cases.
- ✓ Ensure database connectivity and data storage are working properly.
- ✓ Confirm user login and role-based access control.

2.User Training

- ❖ Training Objectives
  - ✓ Enable users to understand system functions.
  - ✓ Teach users how to perform daily hotel operations using the system.
  - ✓ Reduce operational errors and support smooth adoption.
- ❖ Training Target
  - ➢ Customers: Booking rooms and making payments.
  - ➢ Hotel Managers: Viewing reports and monitoring bookings and revenue.
  - ➢ System Administrators : Managing users, rooms, and system configuration.
- ❖ Training Methods
  - ➢ Instructor-led training sessions

- ➢ Hands-on practical demonstrations
- ➢ User manual and quick reference guides
- ➢ Question-and-answer sessions

3.Training Schedule

- ➢ Initial training conducted before system launch.
- ➢ Follow-up training sessions after deployment.
- ➢ Refresher training when system updates are introduced.

4.Post-Installation Support

- ➢ Technical support provided during early system usage.
- ➢ System monitoring to identify and resolve issues
- ➢ Continuous improvement based on user feedback.

5.4 STARTUP STRATEGY

The startup strategy defines the approach for launching and operationalizing the system to ensure a smooth transition from manual or legacy processes to the new computerized system.

1. Objectives of the Startup Strategy

- ✓ Ensure a smooth system deployment and minimize operational disruption.
- ✓ Gradually transition hotel operations from manual or old systems to the new system.
- ✓ Enable all users to adopt the system effectively through proper training and support.
- ✓ Test the system in a controlled environment before full-scale deployment.

2.Startup Phases

A. Pre-Startup Preparation

- ✓ Verify hardware, software, and network infrastructure.
- ✓ Install the system and configure databases and system settings.
- ✓ Create initial user accounts and set access levels.
- ✓ Load initial data such as rooms, pricing, and customer details.

B. Pilot Run

- ✓ Select a small set of users to use the system first.
- ✓ Conduct test operations for bookings, Serve as a reference document for new system
- ✓ Identify potential issues and make necessary adjustments.

C. Full-Scale Deployment

- ✓ Open the system for all hotel operations and users.

- ✓ Monitor system usage Provide on-site support to resolve any issues quickly.

D. Post-Startup

- ✓ Provide continuous technical support and troubleshooting.
- ✓ Collect user feedback for further improvements.
- ✓ Schedule refresher training sessions if needed.

3.Risk management

- ✓ Backup existing data before deployment.
- ✓ Maintain a temporary manual operation for emergencies.
- ✓ Monitor system performance to detect early errors.

4.Success indicators

- ✓ All hotel operations are performed using the system without significant errors.
- ✓ Users can efficiently perform their tasks without reverting to manual processes.
- ✓ System reports and data are accurate and up to date.

CHAPTER SIX : CONCULASION AND RECOMMENDATIONS

6.1 CONCULASION

The Shewa Ber Hotel Management System is designed to improve the efficiency, accuracy, and security of hotel operations. The existing manual system is time-consuming, prone to errors, and difficult to manage. To overcome these challenges, the proposed computerized system provides an integrated solution for managing room bookings, customer information, payments, and administrative tasks.

The system adopts a three-tier software architecture, ensuring separation of concerns, better security, and easier maintenance. By automating hotel processes, the system enhances service quality, improves data management, and supports informed decision making for hotel management.

Overall, the Shewa Ber Hotel Management System meets its objectives by reducing manual work, improving operational efficiency, ensuring data security, and increasing customer satisfaction.

6.2 RECOMMENDATIONS

Based on the analysis and design of the system, the following recommendations are proposed:

1.System Implementation: The hotel should fully implement the proposed computerized system to replace the manual process and improve overall efficiency.

2. User Training: Proper training should be provided to system users, including hotel managers and system administrators, to ensure effective system usage.

3. Data Backup and Security : Regular data backups and strong security measures should be maintained to protect sensitive hotel and customer information.

4. System Maintenance : Continuous system maintenance and updates should be performed to ensure reliability and performance.

5. Future Enhancements :Additional features such as online payment integration, mobile access, and advanced reporting can be added in the future to further improve the system.

6. Monitoring and Evaluation : The system should be regularly monitored and evaluated to identify issues and improve functionality based on user feedback.

6.2 REFERENCE

1.(n.d.). Retrieved from www.businessdictionary.com

2. Agile. (n.d.). http://agilemodeling.com/artifacts/uiPrototype.htm. Retrieved 023,20019, from http://agilemodeling.com/artifacts/uiPrototype.htm

3. google.com. (n.d.).Retrieved from www.google.com

4. hotel management study course. (n.d.). education.com.

5. https:\www.google.come.(n.d.).Retrievedfromwww.google.com.

6. Lausen.s.(2003).IEEE Computer Society,functional requirement.

7. louw, D. (2006 may 10). Description with UML for hotel reservation system

8. Louw, D. (2006). UML for hotel management system.

9. http://www.liacs.nl/assets/abchelorscripties/2006-0&TanneLouw.pdf

10. http://www.itu.dk/~slauesen/Papers/IEEEtasks.pdf.

6.4 APPENDIX

- ✓ Our database query like this

```
CREATE DATABASE Room_Reservation
USE Room_Reservation
CREATE TABLE CUSTOMER (
CUSTOMER_ID INT PRIMARY KEY (CUSTOMER_ID) NOT NULL,
CUSTOMER_FNAME VARCHAR(50) NOT NULL,
CUSTOMER_MNAME VARCHAR(50) NOT NULL,
CUSTOMER_LNAME VARCHAR(50) NOT NULL,
ADD_CITY VARCHAR(50) NOT NULL,
ADD_STATE VARCHAR(50) NOT NULL,
NATIONAL_ID INT NOT NULL UNIQUE,
PHONE_NO VARCHAR(15) NOT NULL,
EMAIL VARCHAR(50) NOT NULL
)

CREATE TABLE SYSTEMADMIN (
ADMIN_ID INT PRIMARY KEY (ADMIN_ID) NOT NULL,
ADMIN_PASSWORD VARCHAR(20) NOT NULL,
ADMIN_USERNAME VARCHAR(20)NOT NULL UNIQUE,
F_NAME VARCHAR(50) NOT NULL,
M_NAME VARCHAR(50)NOT NULL,
L_NAME VARCHAR(50)NOT NULL,
CUSTOMER_ID INT REFERENCES CUSTOMER(CUSTOMER_ID)
)
CREATE TABLE HOTELMANAGER (
MANAGER_ID INT PRIMARY KEY (MANAGER_ID)NOT NULL,
MANAGER_PASSWORD VARCHAR(20) NOT NULL,
MANAGER_USERNAME VARCHAR(20) NOT NULL UNIQUE,
F_NAME VARCHAR(50) NOT NULL ,
M_NAME VARCHAR(50) NOT NULL,
L_NAME VARCHAR(50) NOT NULL,
ADMIN_ID INT REFERENCES SYSTEMADMIN (ADMIN_ID)
)
CREATE TABLE ROOM(
ROOM_ID INT PRIMARY KEY (ROOM_ID) NOT NULL,
ROOM_NO INT NOT NULL UNIQUE,
PRICE  DECIMAL (4,2) NOT NULL ,
TYPE_SINGLE VARCHAR(3) NOT NULL,
TYPE_DOUBLE VARCHAR(3) NOT NULL,
CUSTOMER_ID INT REFERENCES CUSTOMER(CUSTOMER_ID) NOT NULL
)
ALTER TABLE ROOM ALTER COLUMN PRICE DECIMAL(6,2)
CREATE TABLE BOOKING(
BOOK_ID INT PRIMARY KEY (BOOK_ID)NOT NULL,
CHECKIN_DATE DATE NOT NULL,
CHECKOUT_DATE DATE NOT NULL,
CUSTOMER_ID INT REFERENCES CUSTOMER(CUSTOMER_ID) NOT NULL
)
CREATE TABLE PAYMENTGATEWAY(
PAYMENT_ID INT PRIMARY KEY (PAYMENT_ID) NOT NULL,
PAYMENT_DATE DATE  NOT NULL,
PAYMENT_METHOD VARCHAR(20) NOT NULL,
CUSTOMER_ID INT REFERENCES CUSTOMER(CUSTOMER_ID) NOT NULL
)

INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_FNAME,
CUSTOMER_MNAME, CUSTOMER_LNAME, ADD_CITY, ADD_STATE, NATIONAL_ID,
PHONE_NO, EMAIL) VALUES
(1, 'beimnet', 'belay', 'sara', 'Hosanna', 'central ethiopia', 1001, '0993426226',
'beimnet@email.com'),
(2, 'Alemu', 'kasa', 'Smith', 'Addis Ababa', 'Addis', 1002, '0993426354', 'alemusmith@email.com'),
(3, 'Michael', 'Teshe', 'John', 'Hawassa', 'Sidama', 1003, '0461108070', 'michael.j@email.com'),
(4, 'Sara', 'giya', 'gedebo', 'Hosanna', 'central ethiopia', 1004, '0704853473', 'sarah.w@email.com'),
(5, 'David', 'beckham', 'baba', 'Jimma', 'Oromia', 1005, '0972649226', 'david.b@email.com'),
(6,'Abera','Mathiwos','Gada','Wolayita','Southern
ethiopia',1006,'0992238751','sirefaf.@gmail,com'),
(7,'Yosef','Beyene','Minda','Adama','Oromia',1007,'0920990275','dave.@email,com'),
(8,'Beza','Yohannes','Abaynh','Halaba','Central ethiopia',1008,'0938872348','bezi@email.com'),
(9,'Trent','Alexander','Arnold','Toronto','Canada',1009,'+4722345678','trent@gmail.com'),
(10,'Metadel','Tilahun','Tesfaye','Hossana','Central
ethiopia',1010,'0986172397','meti@.gmail.com')

-- Insert into SYSTEMADMIN table
INSERT INTO SYSTEMADMIN (ADMIN_ID, ADMIN_PASSWORD, ADMIN_USERNAME,
F_NAME, M_NAME, L_NAME, CUSTOMER_ID) VALUES
(101, 'ASHA1234', 'ASHUW', 'ASHAGRE', 'WERKICHO', 'KASA', 1)


-- Insert into HOTELMANAGER table
INSERT INTO HOTELMANAGER (MANAGER_ID, MANAGER_PASSWORD,
MANAGER_USERNAME, F_NAME, M_NAME, L_NAME, ADMIN_ID) VALUES
(201, 'GET1234', 'GETSH', 'GETACHEW', 'MEKONIN', 'ZEWDE', 101)

-- Insert into ROOM table
INSERT INTO ROOM (ROOM_ID, ROOM_NO, PRICE, TYPE_SINGLE, TYPE_DOUBLE,
CUSTOMER_ID) VALUES
(301, 101, '800', 'Yes', 'No', 1),
(302, 102, '1200', 'No', 'Yes', 2),
(303, 103, '850', 'Yes', 'No', 3),
(304, 104, '1400', 'No', 'Yes', 4),
(305, 105, '1000', 'Yes', 'No', 5),
(306, 106, '900',  'Yes', 'No', 6),
(307, 107, '1100',  'No',  'Yes',7),
(308, 108, '950' ,  'Yes', 'No',8),
(309, 109,  '1300',  'No' , 'Yes',9),
(310, 110,  '800',   'Yes' ,'No',10);

-- Insert into BOOKING table
INSERT INTO BOOKING (BOOK_ID, CHECKIN_DATE, CHECKOUT_DATE,
CUSTOMER_ID) VALUES
(401, '2025-01-10', '2025-01-15', 1),
(402, '2025-02-01', '2025-02-05', 2),
(403, '2025-03-12', '2025-03-17', 3),
(404, '2025-04-13', '2025-04-18', 4),
(405, '2025-05-14', '2025-05-19', 5 ),
(406, '2025-01-15', '2025-01-20', 6),
(407, '2025-02-16', '2025-02-21', 7),
(408, '2025-03-17', '2025-03-22', 8),
(409, '2025-04-18', '2025-04-23', 9),
(410, '2025-05-19', '2025-05-20', 10 );

-- Insert into PAYMENTGATEWAY table
INSERT INTO PAYMENTGATEWAY (PAYMENT_ID, PAYMENT_DATE,
PAYMENT_METHOD, CUSTOMER_ID) VALUES
(501, '2025-01-09', 'Mobile Banking', 1),
(502, '2025-01-31', 'Internet Banking', 2),
(503, '2025-03-11', 'Cash', 3),
(504, '2025-04-19', 'Mobile Banking', 4),
(505, '2025-05-04', 'Internet Banking', 5),
(506, '2025-06-15', 'Mobile Banking' , 6),
(507, '2025-07-20', 'Internet Banking', 7),
(508, '2025-08-12', 'Mobile Banking' ,8),
(509, '2025-09-18', 'Internet Banking' ,9),
(510, '2025-10-25', 'Mobile Banking' ,10);

 SELECT * FROM PAYMENTGATEWAY
 SELECT * FROM BOOKING
 SELECT * FROM CUSTOMER
 SELECT * FROM ROOM
 SELECT * FROM HOTELMANAGER
 SELECT * FROM SYSTEMADMIN

 ALTER TABLE CUSTOMER ALTER COLUMN CUSTOMER_FNAME VARCHAR(30)
NOT NULL;
 SELECT * FROM CUSTOMER where CUSTOMER_ID =5;
 SELECT * FROM CUSTOMER where CUSTOMER_FNAME='Metadel'and
CUSTOMER_ID=10;
 SELECT * FROM CUSTOMER where PHONE_NO='0992238751';
 SELECT * FROM CUSTOMER where national_id=1007;
 SELECT DISTINCT CUSTOMER_FNAME FROM CUSTOMER;
 SELECT * FROM ROOM WHERE PRICE>1200;
 SELECT * FROM ROOM WHERE ROOM_NO>=105;
 UPDATE CUSTOMER SET CUSTOMER_LNAME='Abebe' where CUSTOMER_ID=8;
 UPDATE CUSTOMER SET CUSTOMER_FNAME='Abenezer' where CUSTOMER_ID=7;
 UPDATE CUSTOMER SET CUSTOMER_LNAME='Debebe' where CUSTOMER_ID=1;
 SELECT * FROM CUSTOMER where CUSTOMER_FNAME like 'M%';
 SELECT * FROM ROOM WHERE PRICE > ANY
 (SELECT PRICE FROM ROOM WHERE PRICE>850);
 SELECT COUNT (*) FROM CUSTOMER;
 SELECT COUNT (*) FROM CUSTOMER WHERE CUSTOMER_ID=7;
 DELETE FROM CUSTOMER WHERE CUSTOMER_LNAME='Abaynh';
 SELECT C.CUSTOMER_ID, C.CUSTOMER_FNAME,C.CUSTOMER_LNAME,
R.ROOM_ID,R.ROOM_NO,PRICE
 FROM CUSTOMER C join ROOM R on C.CUSTOMER_ID=R.CUSTOMER_ID;
 SELECT
C.CUSTOMER_ID,C.CUSTOMER_FNAME,C.CUSTOMER_LNAME,P.PAYMENT_ID,P.PA
YMENT_DATE,P.PAYMENT_METHOD
 FROM CUSTOMER C join PAYMENTGATEWAY P on
C.CUSTOMER_ID=P.CUSTOMER_ID;
 SELECT
C.CUSTOMER_ID,C.CUSTOMER_FNAME,C.CUSTOMER_LNAME,B.BOOK_ID,B.CHEC
KIN_DATE,B.CHECKOUT_DATE
 FROM CUSTOMER C join BOOKING B ON C.CUSTOMER_ID=B.CUSTOMER_ID;
 SELECT * FROM CUSTOMER C left outer join ROOM R on
C.CUSTOMER_ID=R.CUSTOMER_ID;
 SELECT * FROM CUSTOMER C right outer join ROOM R on
C.CUSTOMER_ID=R.CUSTOMER_ID;
 SELECT * FROM CUSTOMER,ROOM SELECT * FROM CUSTOMER C full outer join
ROOM R on C.CUSTOMER_ID=R.CUSTOMER_ID;
 SELECT * FROM CUSTOMER C left outer join PAYMENTGATEWAY P on
C.CUSTOMER_ID=P.CUSTOMER_ID;
  SELECT * FROM CUSTOMER C right outer join PAYMENTGATEWAY P on
C.CUSTOMER_ID=P.CUSTOMER_ID;
SELECT * FROM CUSTOMER,PAYMENTGATEWAY SELECT * FROM CUSTOMER C
FULL outer join PAYMENTGATEWAY P on C.CUSTOMER_ID=P.CUSTOMER_ID;
 SELECT * FROM CUSTOMER C left outer join BOOKING B on
C.CUSTOMER_ID=B.CUSTOMER_ID;
 SELECT * FROM CUSTOMER C right outer join BOOKING B on
C.CUSTOMER_ID=B.CUSTOMER_ID;
 SELECT * FROM CUSTOMER,BOOKING SELECT * FROM CUSTOMER C full outer join
BOOKING B on C.CUSTOMER_ID=B.CUSTOMER_ID;
```