B.M.S. COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

Lab Record

Object-Oriented Modeling – 23CS5PCOOM

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of
Engineering in
Computer Science and Engineering

Submitted by:

Dyuthi

1BM23CS097

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
August 2025-December 2025

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Object-Oriented Modeling(23CS5PCOOM) laboratory has been carried out by Dyuthi (1BM23CS097) during the 5th Semester August 2025-December 2025

.

Signature of the Faculty Incharge:
Sonika Sharma
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

1.Hotel Management System

Problem Statement

Hotels face challenges in managing reservations, guest check-in/check-out, room availability, billing, and staff operations due to manual or outdated processes. These inefficiencies lead to booking errors, reduced guest satisfaction, and operational delays. A centralized, automated Hotel Management System is required to streamline operations, improve accuracy, and enhance customer service.

Software Requirements Specification (SRS)

1. Introduction

The Hotel Management System is designed to automate hotel operations and reduce manual workload across departments. It supports reservation handling, room management, guest records, and billing tasks. The system ensures faster service, improved accuracy, and better customer satisfaction. It aims to create a smooth workflow for receptionists, managers, and accountants.

2. Overall Description

The system provides integrated modules for booking, check-in/check-out, staff coordination, and payment processing. It maintains a centralized database to keep track of room status and guest information in real time. Users interact with simple interfaces designed to speed up daily tasks. It helps management monitor performance through automatically generated reports.

3. External Interface Requirements

The system features a graphical interface for hotel staff, supports printers to generate invoices, and interacts with the hotel's internal database for data storage. It may optionally connect to SMS/email services for customer notifications.

4. System Features (Functional Requirements)

- FR1: The system shall allow staff to create, modify, and cancel room reservations.
- FR2: The system shall manage check-in and check-out activities with automatic updates to room status.
- FR3: The system shall generate bills and process payments.
- FR4: The system shall store and update customer profiles.
- FR5: The system shall allocate rooms based on availability and booking type.

- FR6: The system shall maintain staff schedules and housekeeping status.
- FR7: The system shall generate daily, weekly, and monthly activity reports.
- FR8: The system shall allow managers to view occupancy and revenue statistics.

5. Non-Functional Requirements

- NFR1: The system shall ensure secure access through role-based authentication.
- NFR2: The system shall respond to user actions within 2–3 seconds.
- NFR3: The system shall maintain accurate and consistent data across all modules.
- NFR4: The system shall include automatic backups of hotel records.
- NFR5: The interface shall be simple and easy for new staff to learn.
- NFR6: The system shall have high availability during business hours.
- NFR7: The system shall run on standard hardware with minimal resource usage.
- NFR8: The system shall protect sensitive customer information using encryption techniques.

Class Diagram:

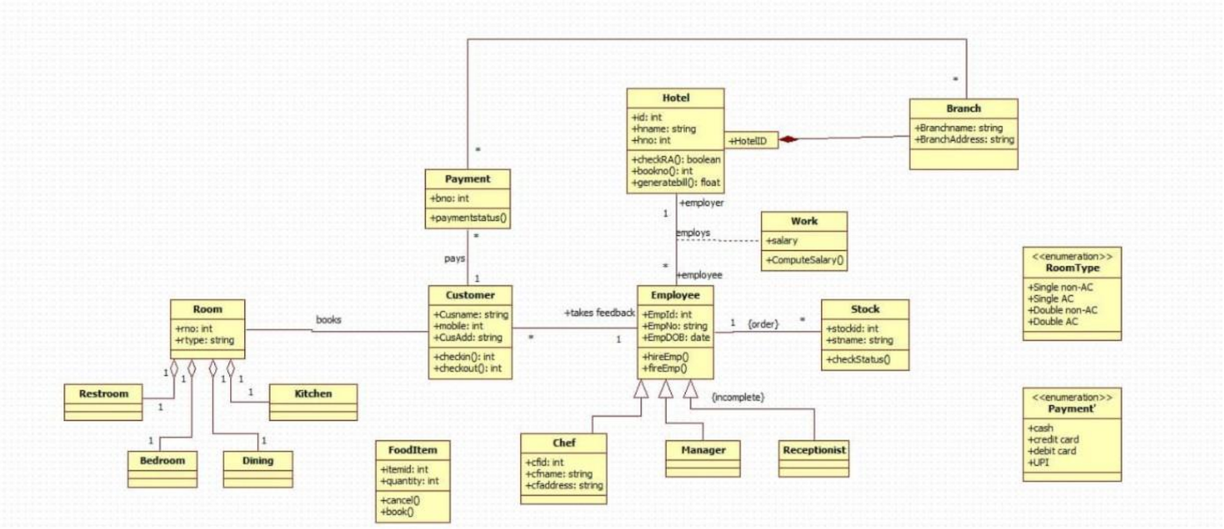

Fig 1.1 Class diagram of Hotel Management System

The diagram shows a Hotel Management System where a hotel has multiple branches, rooms, customers, employees, and payments. Customers book rooms and make payments. Employees (like chefs, managers, and receptionists) work for the hotel and handle tasks such as food orders and customer service. Rooms have different types and facilities, while stock and food items are also managed within the system.

State Diagram:

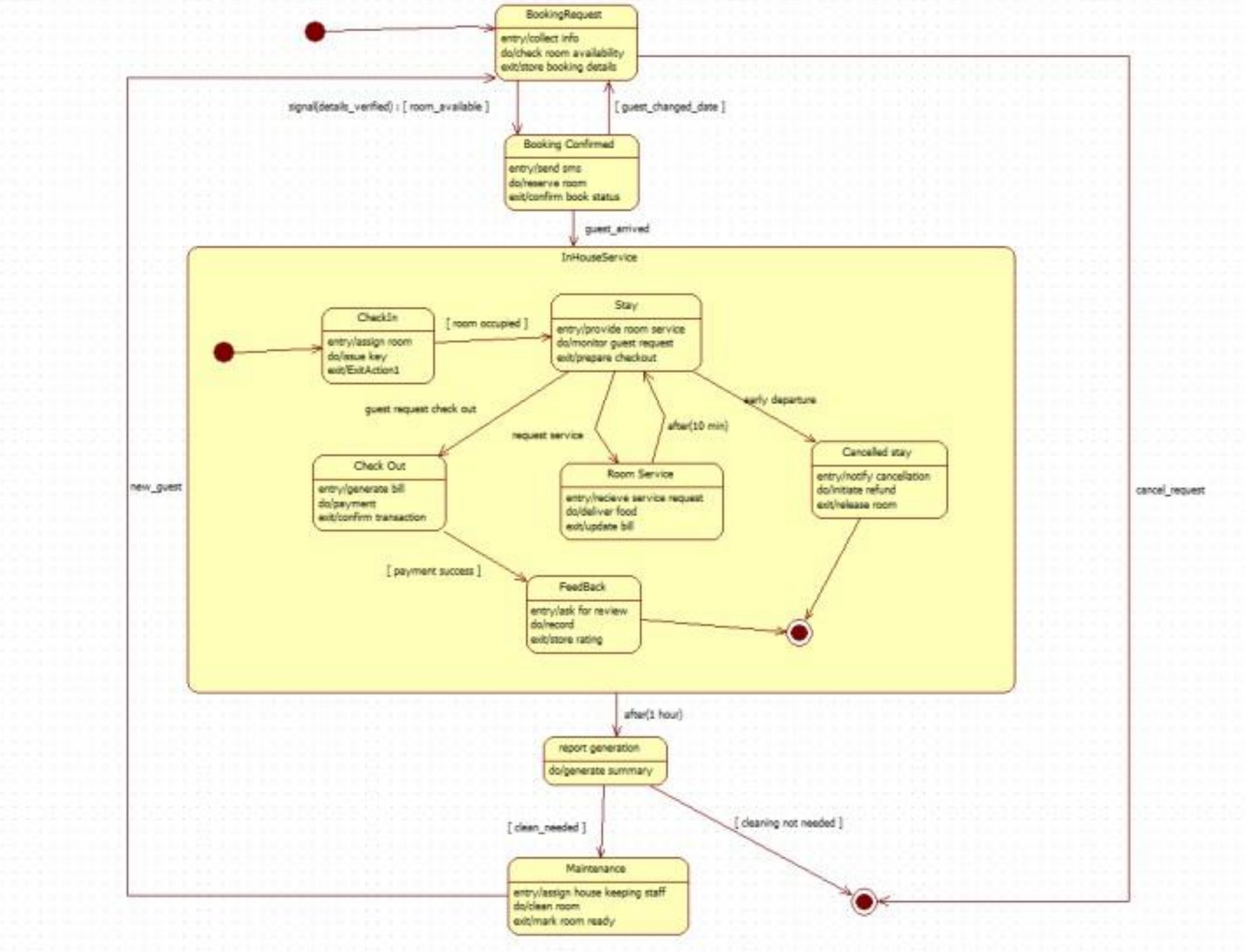

Fig 1.2 State diagram of Hotel Management System

The state diagram shows the lifecycle of a hotel booking and stay. It begins with a Booking Request, which becomes Booking Confirmed if a room is available. When the guest arrives, they move through states such as Check-in, Stay, requesting Room Service, giving Feedback, and finally Check-out. A guest may also enter a Cancelled Stay if the booking is canceled. After checkout, the system generates reports, and rooms may go into Maintenance if cleaning is needed.

Use-Case Diagram:

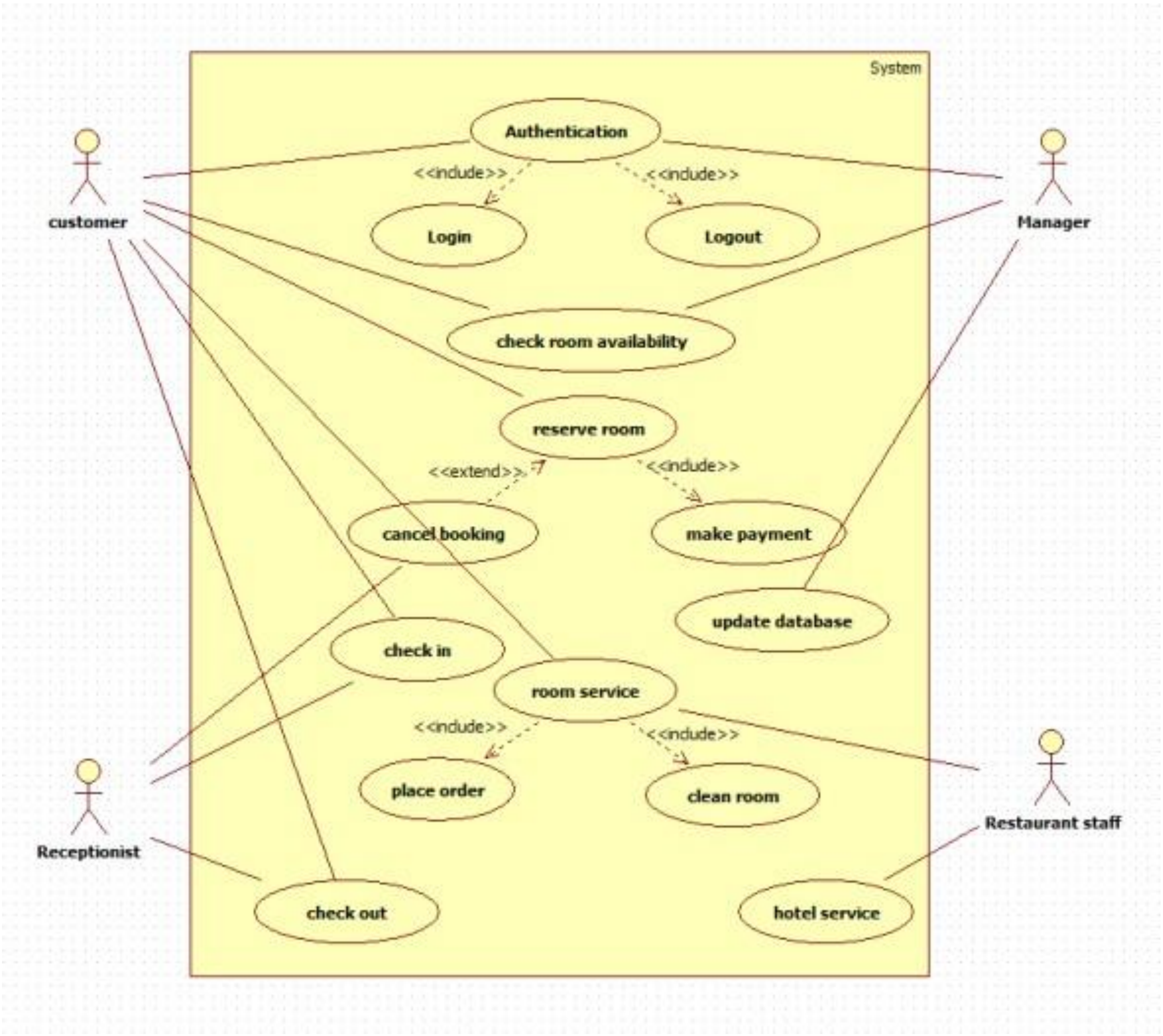

Fig 1.3 Use case diagram of Hotel Management System

The use-case diagram shows how different users interact with the Hotel Management System. Customers can log in, check room availability, reserve or cancel bookings, make payments, check in, request room service, and check out. Receptionists handle check-ins, check-outs, and room service tasks like placing orders or cleaning rooms. Managers can authenticate, update the database, and manage reservations. Restaurant staff provide hotel and food service support. The system centralizes all these actions to streamline hotel operations

Sequence Diagram:

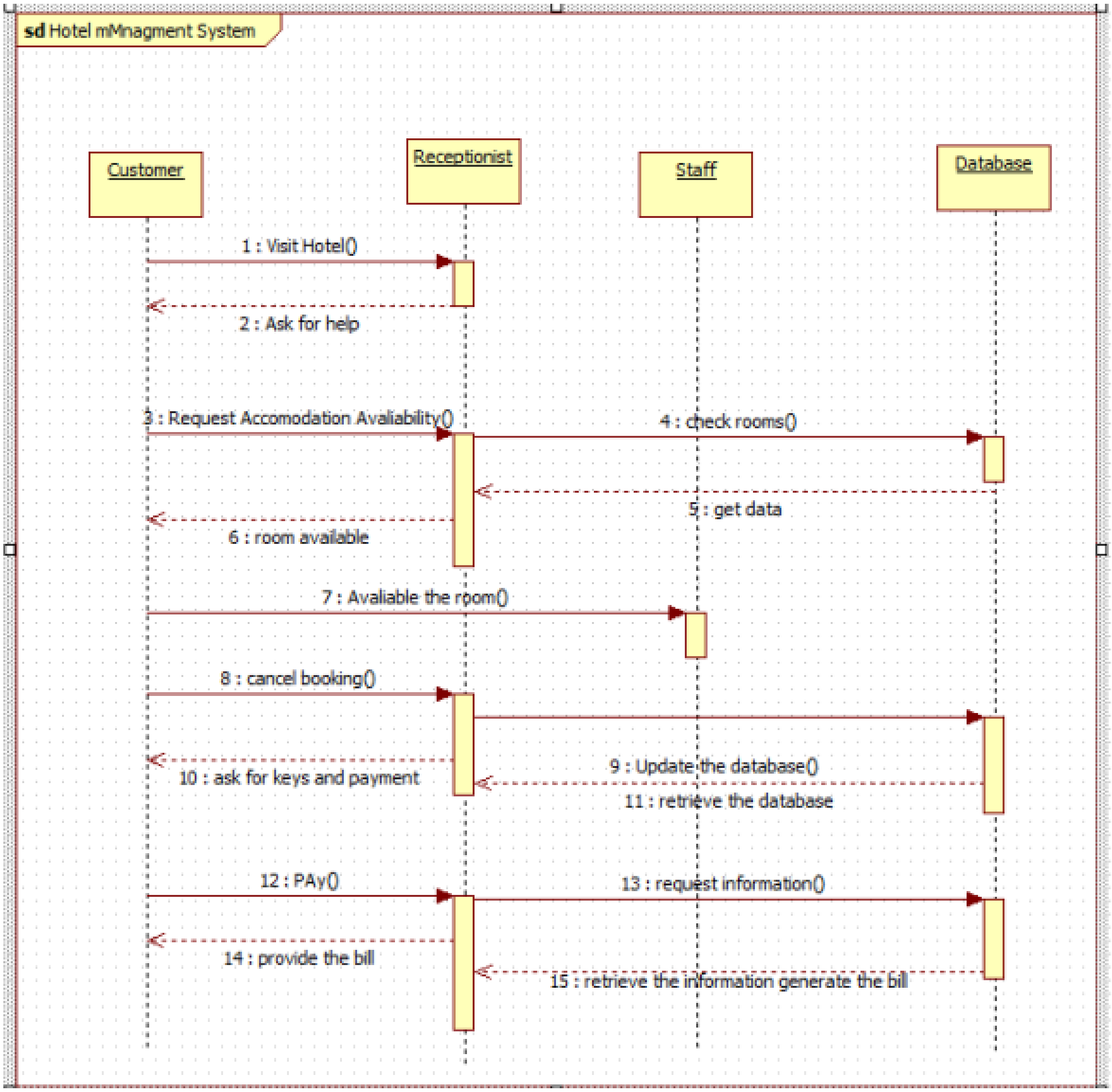
sd Hotel mMnagment System
Customer
Receptionist
Staff
Database
1 : Visit Hotel()
2 : Ask for help
3 : Request Accomodation Avaliability()
4 : check rooms()
5 : get data
6 : room available
7 : Avaliable the room()
8 : cancel booking()
9 : Update the database()
10 : ask for keys and payment
11 : retrieve the database
12 : PAy()
13 : request information()
14 : provide the bill
15 : retrieve the information generate the bill

Fig 1.4 Sequence diagram of Hotel Management System

The sequence diagram shows the interaction between the customer, receptionist, staff, and database during a hotel booking process. The customer visits the hotel, requests accommodation, and the receptionist checks room availability through the staff and database. Once availability is confirmed, the customer can proceed to book or cancel. The receptionist updates the database, collects payment, and requests billing information. Finally, the receptionist provides the bill to the customer.

Activity diagram:

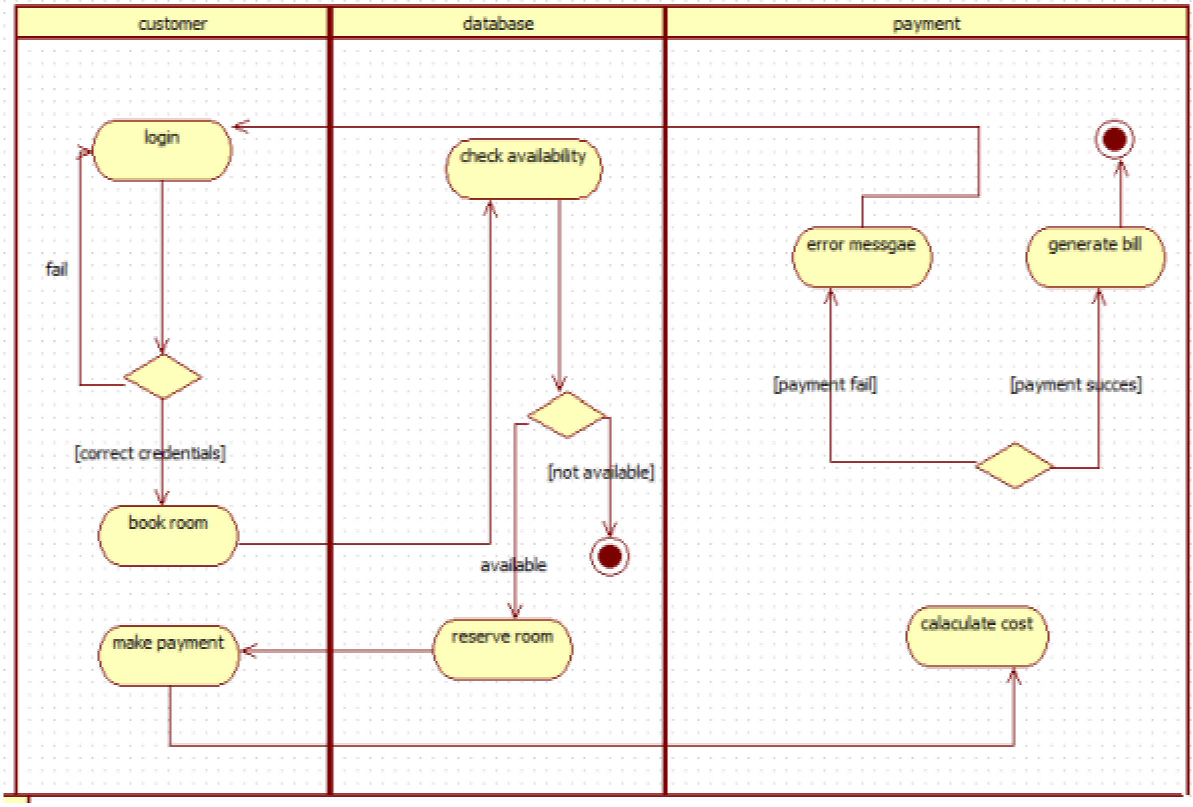

Fig 1.5 Activity diagram of Hotel Management System

The activity diagram shows the steps a customer follows to book a room in the hotel system. The customer logs in, and the database checks room availability. If a room is available, the customer proceeds to book and make a payment. The payment system calculates the cost and either generates the bill for a successful payment or shows an error message if the payment fails.

2. Credit Card Automation System

Problem Statement

Manual credit card processing—including application approval, transaction handling, billing, payments, and fraud detection—leads to delays, errors, and security risks. As customers and transactions increase, managing these tasks becomes more difficult for banks.

A Credit Card Automation System is required to automate account creation, real-time transaction processing, billing cycles, payment updates, fraud alerts, and reporting. The goal is to improve speed, accuracy, security, and overall efficiency through a centralized, secure, and user-friendly platform.

Software Requirements Specification (SRS)

1. Introduction

The Credit Card Processing System provides secure authorization and settlement of card transactions. It aims to ensure fast, reliable payment handling for merchants and customers. The system validates card details, checks available credit, and processes transactions accurately. It reduces human error and prevents fraudulent activity.

2. Overall Description

The system connects with banking networks to verify customer accounts and approve or decline payments. It maintains complete transaction histories for accountability and auditing. It provides interfaces for merchants, bank staff, and system administrators. The system focuses on real-time processing and high-security communication.

3. External Interface Requirements

The system interacts with card readers, merchant terminals, and banking servers. It provides a clean UI for merchants to view transaction results. It uses secure network protocols for communication with financial institutions.

4. System Features (Functional Requirements)

- FR1: The system shall validate card number, expiry date, and CVV.
- FR2: The system shall check available credit before approving a transaction.
- FR3: The system shall authorize or decline payments based on bank response.
- FR4: The system shall record every transaction in the system database.
- FR5: The system shall generate receipts for approved transactions.
- FR6: The system shall flag suspicious or unusual activity.

- FR7: The system shall support refunds and chargebacks.
- FR8: The system shall provide administrators with transaction summaries.

5. Non-Functional Requirements

- NFR1: The system shall ensure strong data encryption for all communications.
- NFR2: Transactions shall be processed within 2–3 seconds.
- NFR3: The system shall operate 24/7 with minimal downtime.
- NFR4: The system shall comply with financial security standards (e.g., PCI-DSS).
- NFR5: The system shall ensure high reliability and error-free performance.
- NFR6: The interface shall be simple and readable for merchants.
- NFR7: The system shall provide logs for all critical operations.
- NFR8: The system shall handle large transaction volumes without slowdown.

Class Diagram:

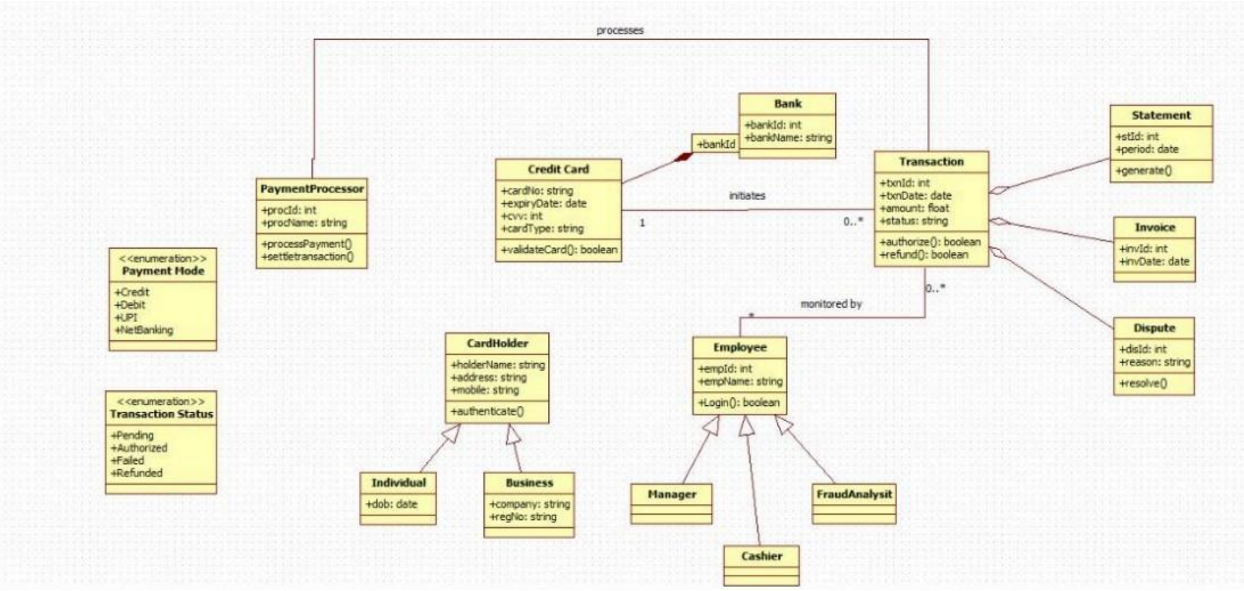

Fig 2.1 Class diagram of Credit Card Processing System

The class diagram represents a payment processing system that manages credit card transactions. A PaymentProcessor handles payment modes (like credit, debit, UPI, net banking) and interacts with CreditCard details, which are validated before initiating a transaction. Each Transaction is linked to a Bank and can generate related documents such as Statements, Invoices, or Disputes. Transactions carry statuses like pending, authorized, failed, or refunded.

A CardHolder (either an individual or business) owns the credit card and can authenticate themselves. Employees—including managers, cashiers, and fraud analysts—monitor transactions and log into the system. Overall, the diagram shows how payments flow from cardholder verification to bank interaction, transaction processing, and post-transaction documentation.

State diagram:

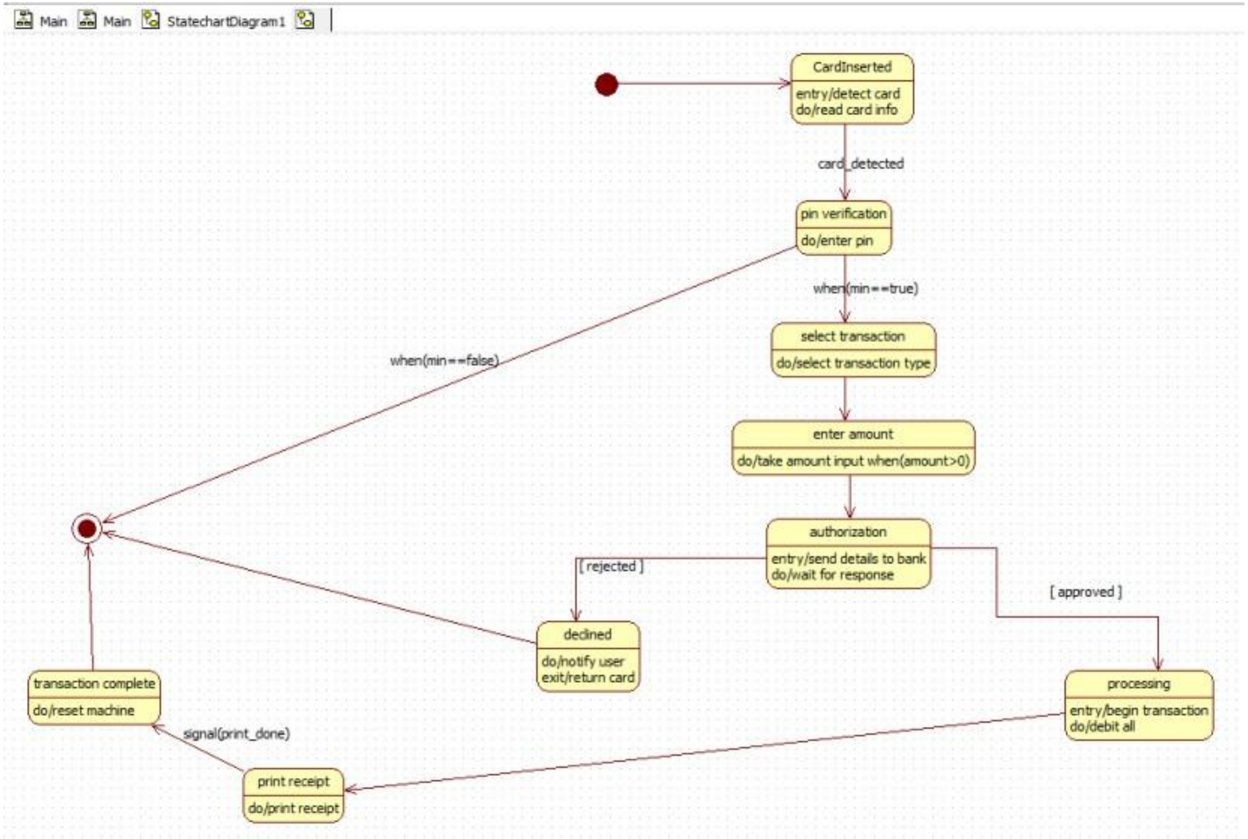

Fig 2.2 State diagram of Credit Card Processing System

The state diagram illustrates the workflow of a card-based transaction process in a payment machine (like an ATM or POS). The process begins when a card is inserted and detected, followed by PIN verification. If the PIN is valid, the user selects a transaction type and enters an amount. The machine then sends the transaction details to the bank for authorization.

If the bank approves the request, the system proceeds to process the transaction and later prints a receipt. If the request is rejected, the machine notifies the user and returns the card. In both scenarios, the process ends with the machine resetting itself and returning to the initial state, ready for the next transaction.

Use-case diagram:

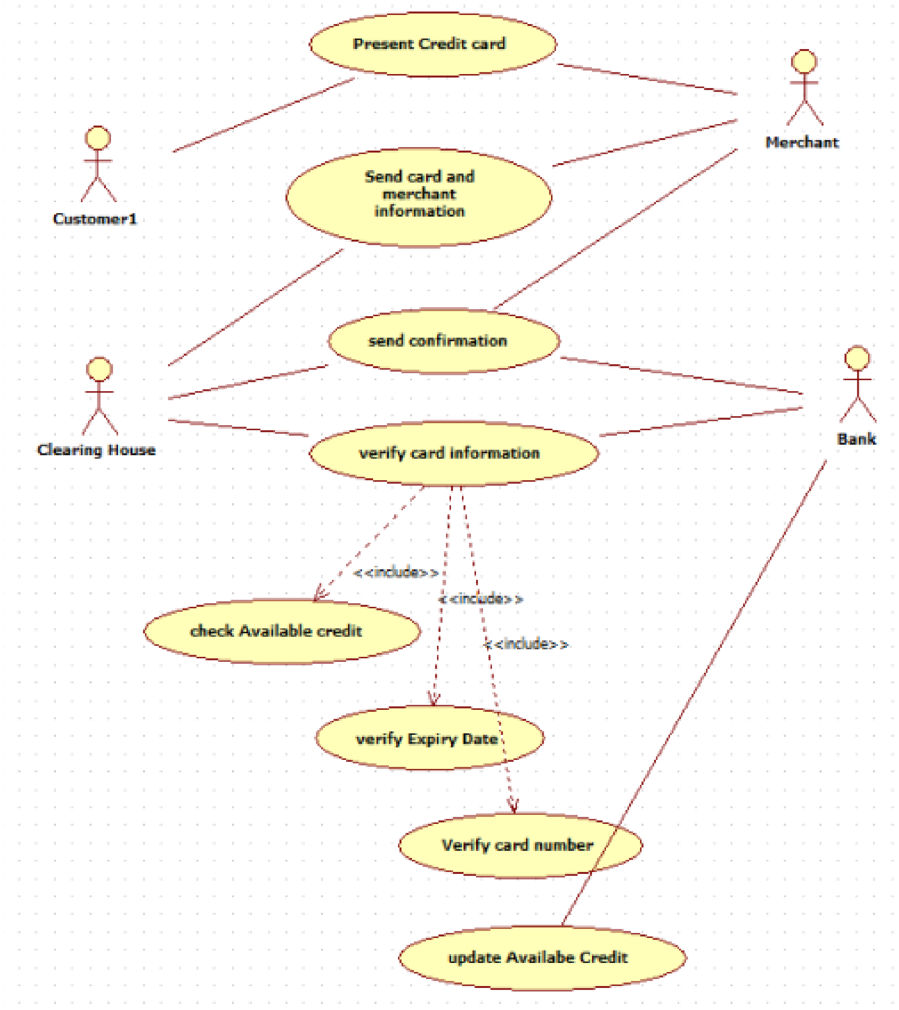

Fig 2.3 Use-case diagram of Credit Card Processing System

The use case diagram illustrates the overall credit card processing workflow involving the Customer, Merchant, Clearing House, and Bank. The process begins when the customer presents their credit card to the merchant, who sends the card and merchant details for verification. The Clearing House and Bank validate the card through a series of checks, including verifying the card number, expiry date, and available credit.After successful verification, the system updates the available credit and sends a confirmation back to the merchant. This enables the merchant to complete the transaction. The diagram highlights the interaction between different actors and the essential verification steps involved in processing a credit card payment.

Sequence diagram:

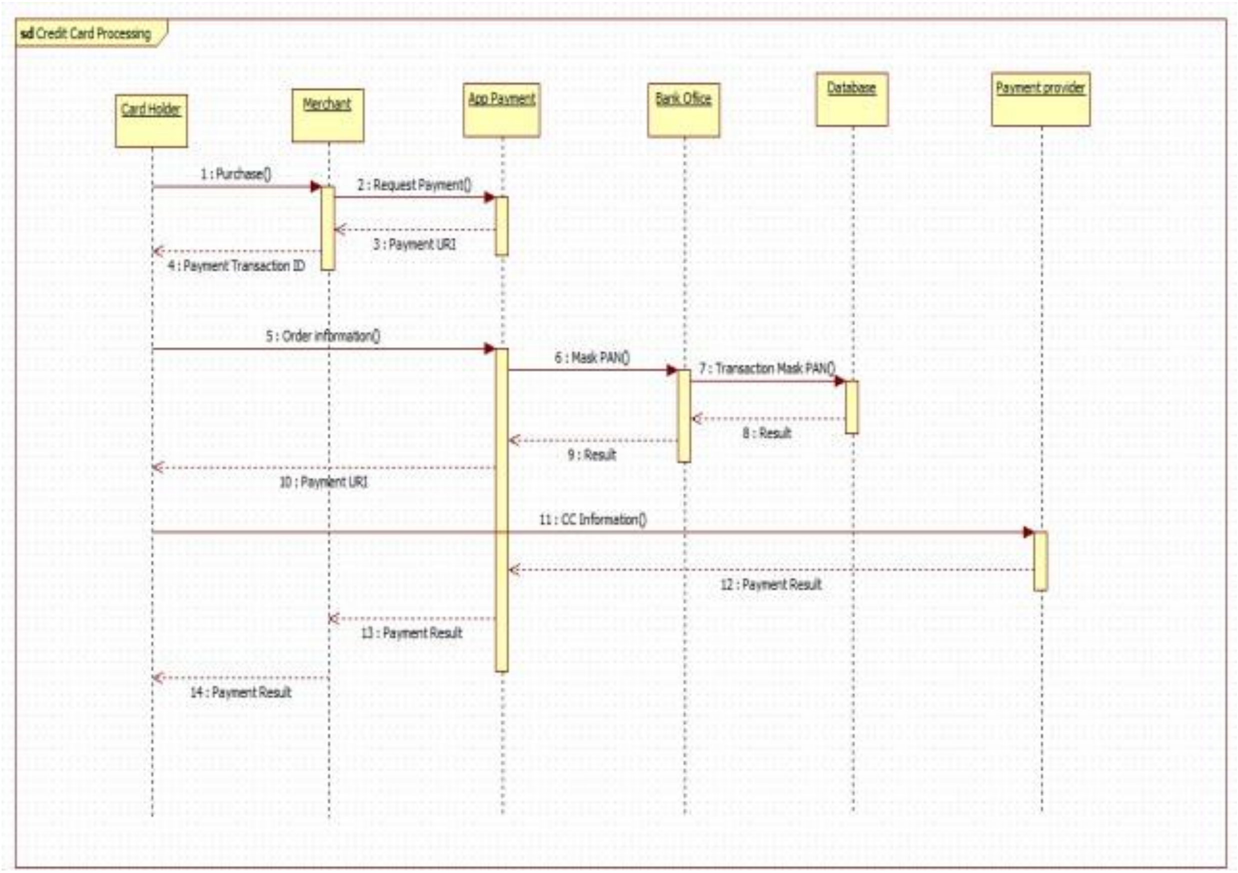

Fig 2.4 Sequence diagram of Credit Card Processing System

The sequence diagram shows the flow of a credit card payment from purchase to completion. The Card Holder initiates a payment, the Merchant sends the request, and the App Payment system processes the order by masking card details and communicating with the Bank Office and Database for verification. After validation, the payment result is returned through the App Payment system to the Merchant, Payment Provider, and finally to the Card Holder.

Activity diagram:

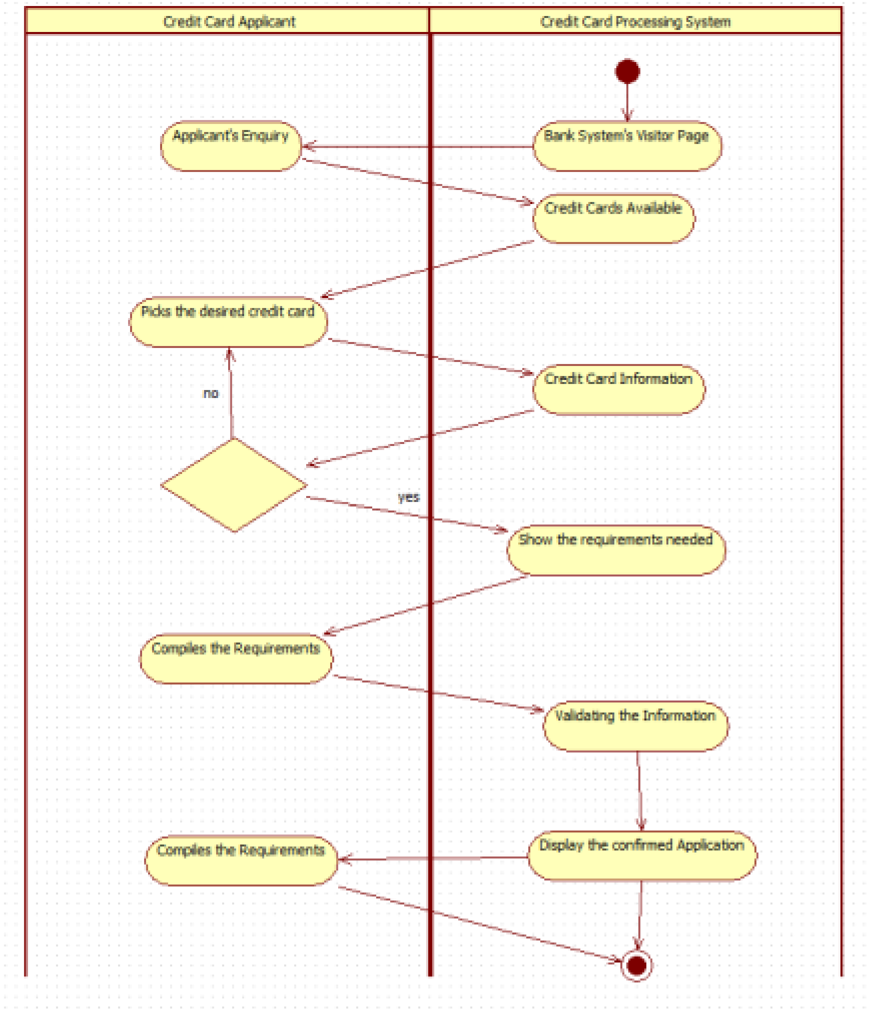
Credit Card Applicant
Credit Card Processing System
Bank System's Visitor Page
Applicant's Enquiry
Credit Cards Available
Picks the desired credit card
Credit Card Information
no
yes
Show the requirements needed
Compiles the Requirements
Validating the Information
Display the confirmed Application
Compiles the Requirements

Fig 2.5 Activity diagram of Credit Card Processing System

The activity diagram illustrates the process of a credit card applicant applying through the bank's credit card processing system. The applicant sends an enquiry, views available credit cards, and selects one. The system displays the card information and required documents. After the applicant submits the necessary requirements, the system validates the information and finally displays the confirmed application.

3. Library Management System

Problem Statement

Traditional library operations such as book issuing, returning, cataloging, fine calculation, and record maintenance are often done manually, leading to errors, delays, and difficulty in tracking books and users. As the number of books and members increases, managing inventory, reducing book loss, and maintaining accurate records becomes challenging.

A Library Management System is needed to automate book management, user registration, transactions, and report generation. The goal is to improve efficiency, accuracy, and accessibility for librarians and users. The system should provide quick search, real-time inventory updates, and role-based access while ensuring secure data handling.

Software Requirements Specification (SRS)

1. Introduction

The Library Management System is designed to streamline daily library operations such as book cataloging, issuing, returning, and member management. It eliminates manual record-keeping and reduces common errors. The system helps librarians maintain accurate inventories and provides members with easy access to available books. It improves efficiency, transparency, and overall service quality.

2. Overall Description

The system maintains detailed records of books, members, borrowed items, and overdue fines. It provides separate interfaces for librarians and members for easy interaction. A centralized database keeps book availability updated in real time. It supports quick searching, tracking, and reporting to make library operations faster and more organized.

3. External Interface Requirements

The system includes graphical interfaces for librarians and students, supports barcode scanners for book identification, and connects to a backend database. It may also send email reminders about due dates or overdue books.

4. System Features (Functional Requirements)

- FR1: The system shall allow librarians to add, update, and delete book records.
- FR2: The system shall allow members to search for books by title, author, or category.
- FR3: The system shall manage book issue and return transactions.

- FR4: The system shall update book availability automatically after each transaction.
- FR5: The system shall calculate and record overdue fines.
- FR6: The system shall maintain member profiles and borrowing history.
- FR7: The system shall generate lists of overdue books and fines.
- FR8: The system shall generate monthly library activity reports.

5. Non-Functional Requirements

- NFR1: The system shall ensure data accuracy and prevent duplication of book entries.
- NFR2: It shall maintain secure login access for librarians and members.
- NFR3: The interface shall be simple, intuitive, and suitable for frequent use.
- NFR4: The system shall process search queries within 1–2 seconds.
- NFR5: It shall be reliable and available during library operating hours.
- NFR6: The system shall support backup and recovery of book data.
- NFR7: The system shall protect member information from unauthorized access.
- NFR8: It shall work smoothly on standard library computers with minimal resources.

Class diagram:

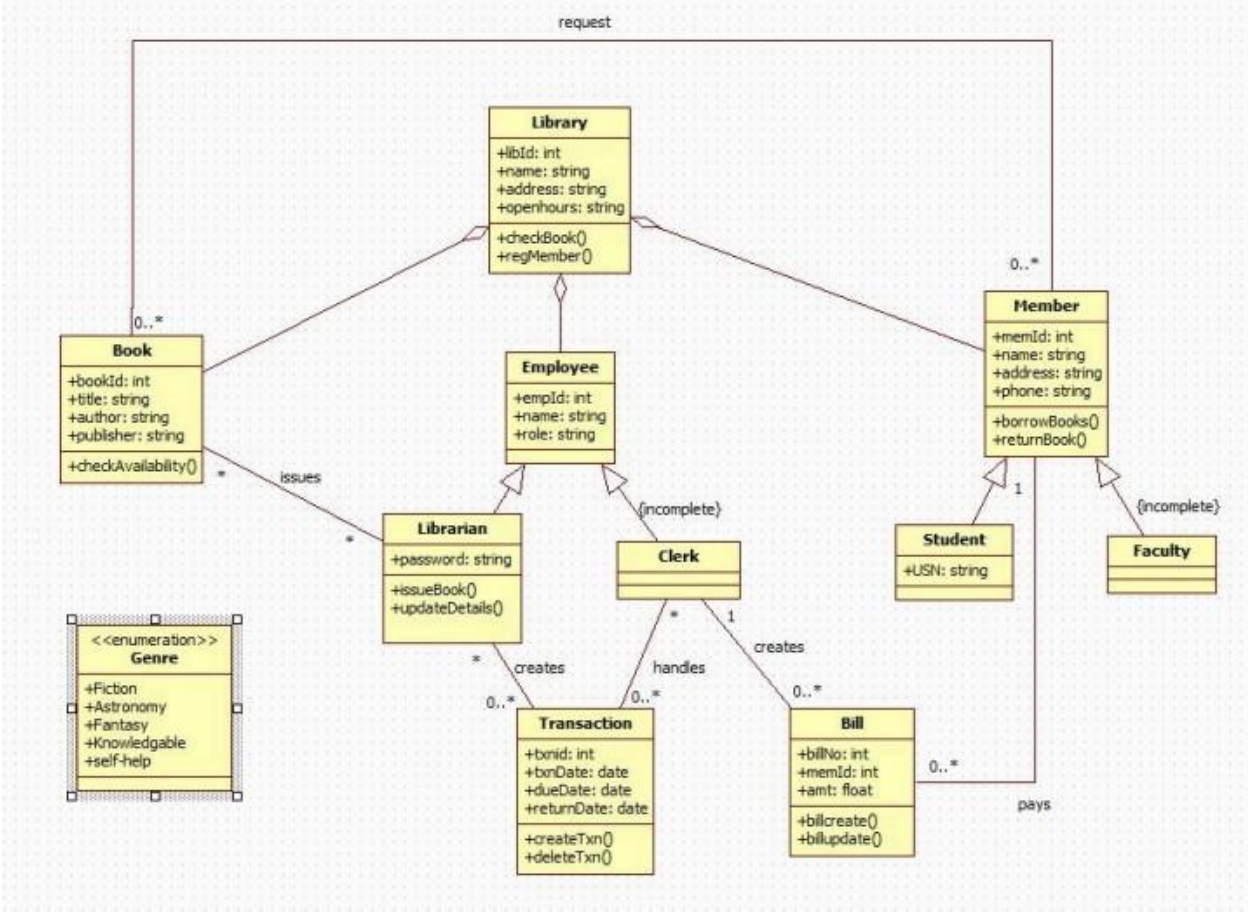

Fig 3.1 Class diagram of Library management system

The class diagram represents a library management system involving books, members, employees, and transactions. The Library maintains books and registered members. Members (students or faculty) can borrow and return books, while employees (librarians and clerks) manage book issuing, record updates, and transaction handling. Each transaction records borrowing and return dates, and bills are generated for any dues. The diagram also includes book genres and shows how different roles interact to support the overall library operations.

State diagram:

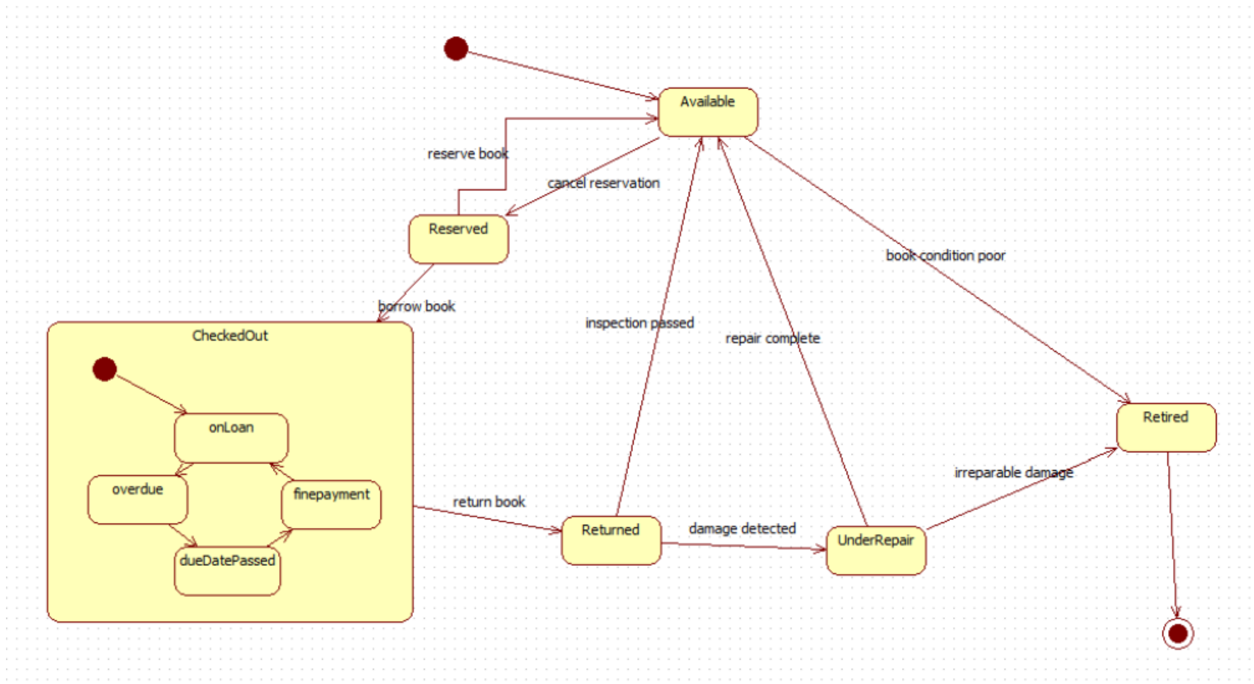

Fig 3.2 State diagram of Library management system

The state diagram shows the lifecycle of a library book. A book starts in the Available state and can be Reserved or Checked Out. When checked out, it may become overdue or require fine payment before return. Once returned, the book is inspected—if damaged, it goes Under Repair; if repair is completed, it becomes available again. If the book is irreparably damaged or in poor condition, it transitions to the Retired state and is removed from circulation.

Use-case diagram:

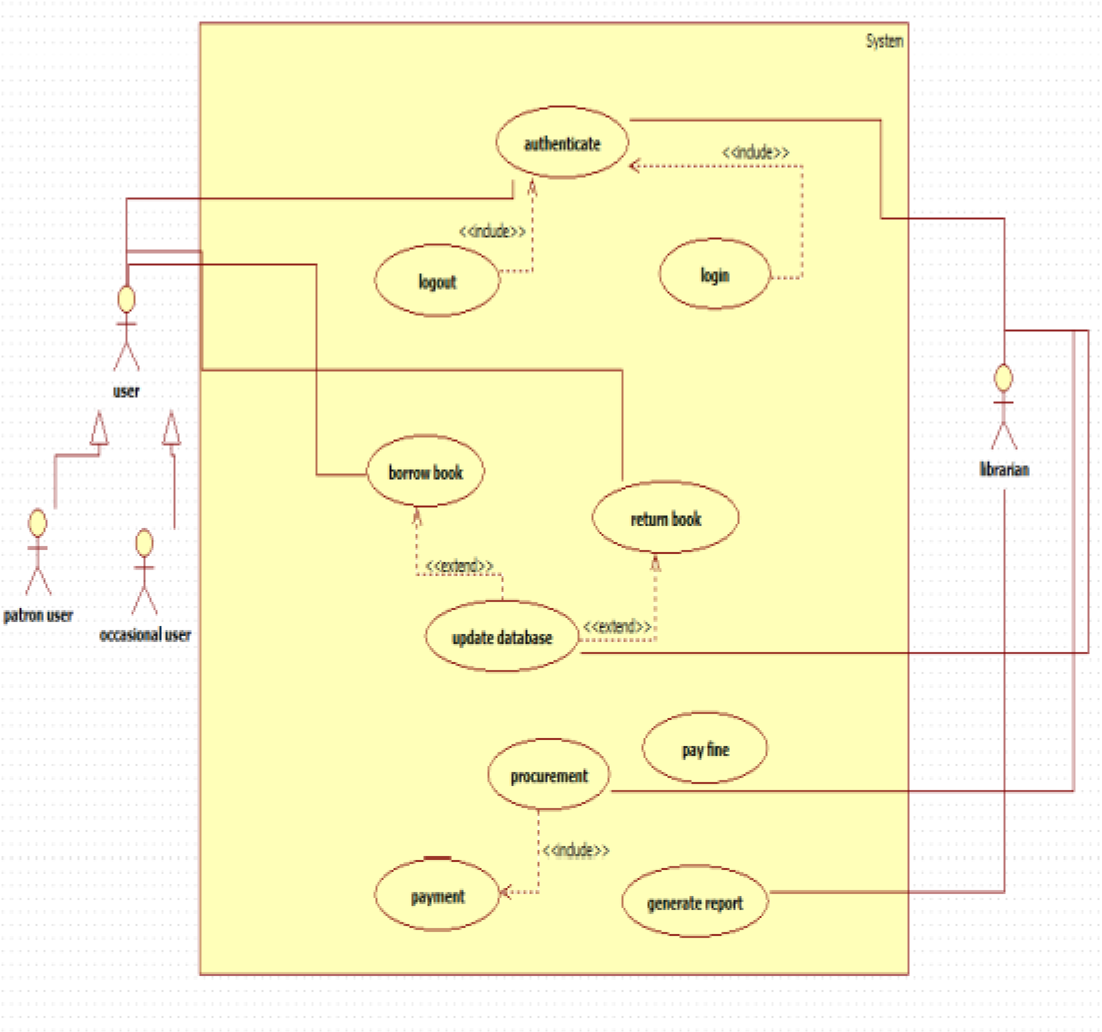

Fig 3.3 Use case diagram of Library management system

The use case diagram represents a library management system where users (patron and occasional users) and librarians interact with various system functions. Users can authenticate, log in, borrow books, return books, pay fines, and make payments. Librarians manage procurement, generate reports, and update the database. Several use cases include or extend others, showing how core actions like authentication and data updates support the overall library operations.

Sequence diagram:

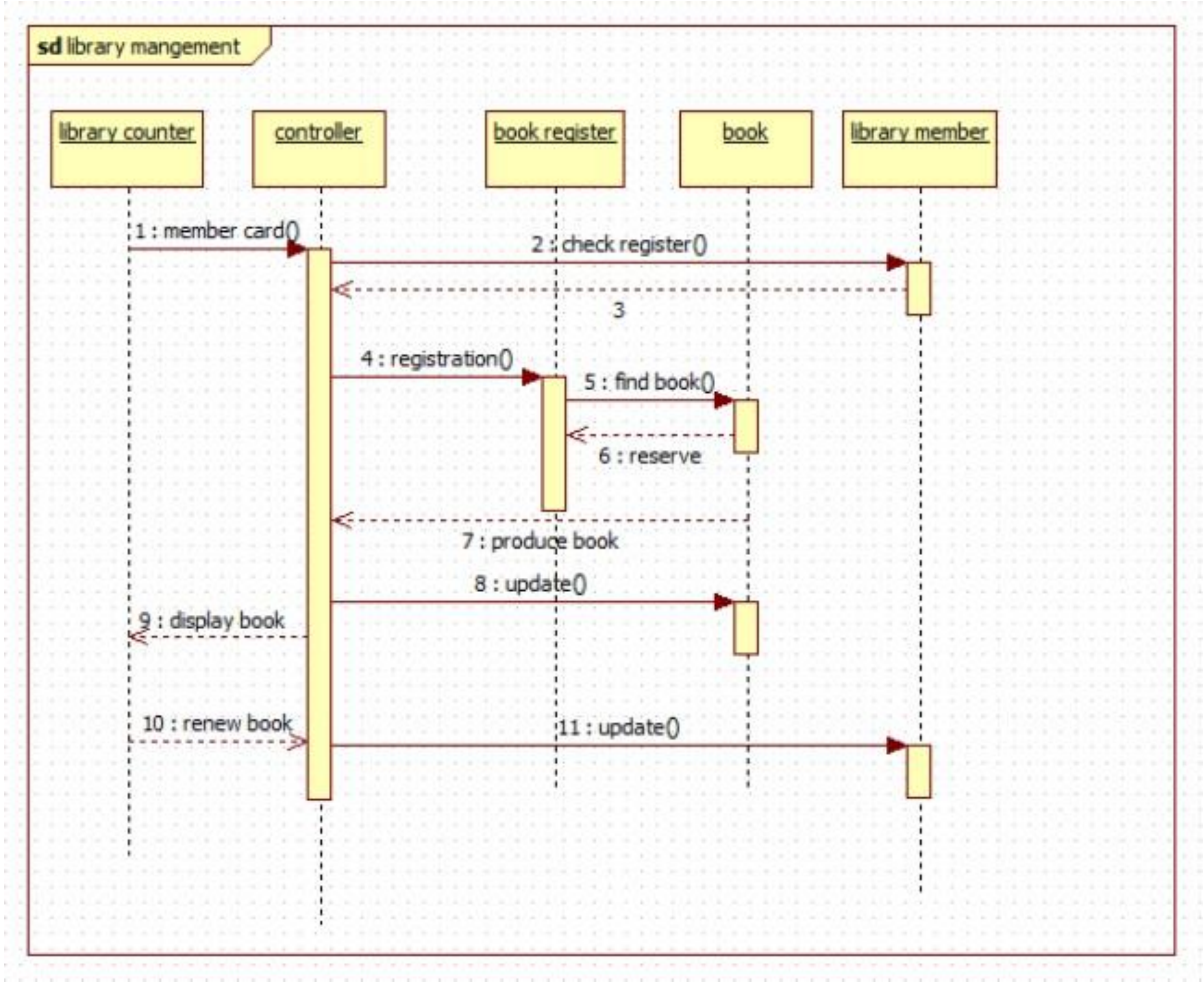

Fig 3.4 Sequence diagram of Library management system

The sequence diagram shows the process of managing book transactions in a library. A library member presents their card at the library counter, which triggers the controller to check the book register, register the member, and search for the requested book. Once the book is found and reserved, it is produced for the member. The system updates the book's status, displays it to the member, and also handles book renewal requests, updating the records accordingly.

Activity diagram:

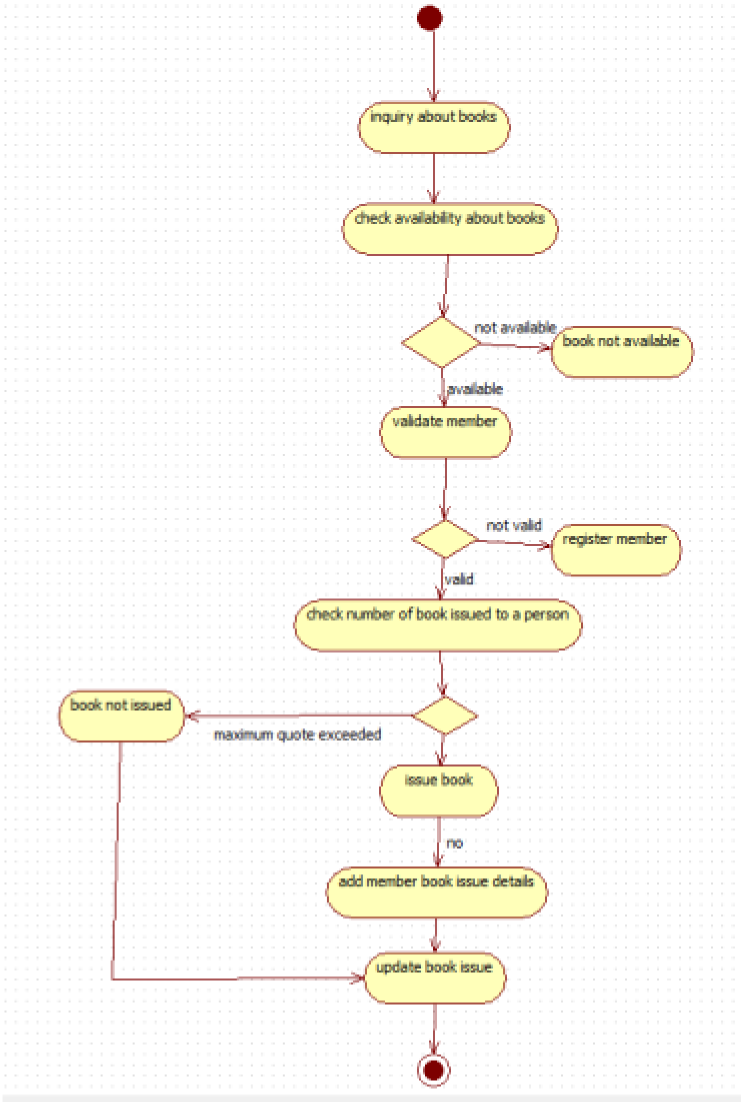

Fig 3.5 Activity diagram of Library management system

The activity diagram outlines the process of issuing a book in a library. It begins with a user inquiring about a book and checking its availability. If available, the system validates the member; if not registered, the member is asked to register. The system then checks whether the member has exceeded the maximum number of issued books. If eligible, the book is issued, the member's issue details are added, and the book issue record is updated. Otherwise, the book is not issued.

4. Stock Maintenance System

Problem Statement

Many businesses still rely on manual methods to record and track stock levels, purchases, and sales. This often leads to errors, mismatched inventory counts, delays in restocking, and difficulty in monitoring product movement. As stock increases, it becomes harder to maintain accurate records and prevent shortages or overstocking.

A Stock Maintenance System is needed to automate inventory tracking, update stock levels in real time, and generate reports on stock usage, shortages, and reorder requirements. The goal is to ensure accuracy, reduce manual errors, improve efficiency, and help businesses make better inventory decisions through a centralized and user-friendly system.

Software Requirements Specification (SRS)

1. Introduction

The Stock Maintenance System is designed to improve the efficiency of inventory management by monitoring product availability, supplier details, stock movement, and purchase transactions. The system minimizes manual errors and ensures that accurate stock records are maintained at all times. It supports better planning and decision-making by providing real-time stock levels to the management. The system ultimately helps organizations reduce wastage, prevent shortages, and maintain smooth operational flow.

2. Overall Description

The system stores detailed information about each product, including quantity, category, supplier, and purchase price. It monitors stock inflow and outflow, updating quantities automatically after every transaction. Users can categorize items for easier tracking and generate inventory reports as needed. The system provides alerts when items fall below predefined minimum levels, ensuring timely restocking. It also maintains complete logs of stock adjustments for auditing and transparency.

3. External Interface Requirements

The system provides a user-friendly graphical interface for inventory managers, supports barcode scanners for quick product identification, and interacts with a backend database for storing all stock details. It can export reports to PDF or Excel and may integrate with billing or sales systems in the future. Optional support for printers is available for generating physical inventory reports.

4. System Features (Functional Requirements)

- FR1: The system shall allow users to add, update, search, and delete product records.
- FR2: The system shall automatically update stock quantities after each purchase or issue.
- FR3: The system shall provide minimum stock alerts to avoid shortages.
- FR4: The system shall manage supplier profiles and contact information.

- FR5: The system shall store purchase orders and maintain complete transaction history.
- FR6: The system shall record stock adjustments made by authorized users.
- FR7: The system shall generate daily, weekly, and monthly inventory reports.
- FR8: The system shall support product categorization for bulk management.
- FR9: The system shall allow managers to view stock value calculations for auditing.
- FR10: The system shall maintain a log of deleted or modified items for accountability.

5. Non-Functional Requirements

- NFR1: The system shall maintain accurate, error-free, and consistent stock records at all times.
- NFR2: It shall respond to stock update and search operations within 2–3 seconds.
- NFR3: The interface shall be intuitive, clean, and usable without extensive training.
- NFR4: The system shall ensure secure access through role-based authentication.
- NFR5: It shall remain reliable during constant use with minimal downtime.
- NFR6: Data backups shall be automatically scheduled to prevent loss during failures.
- NFR7: The system shall ensure that only authorized personnel can modify or delete stock entries.
- NFR8: It shall run efficiently on standard business hardware and support future scalability.
- NFR9: The system shall protect sensitive business data through encryption.
- NFR10: The system shall maintain high compatibility with printers and external report formats.

Class diagram:

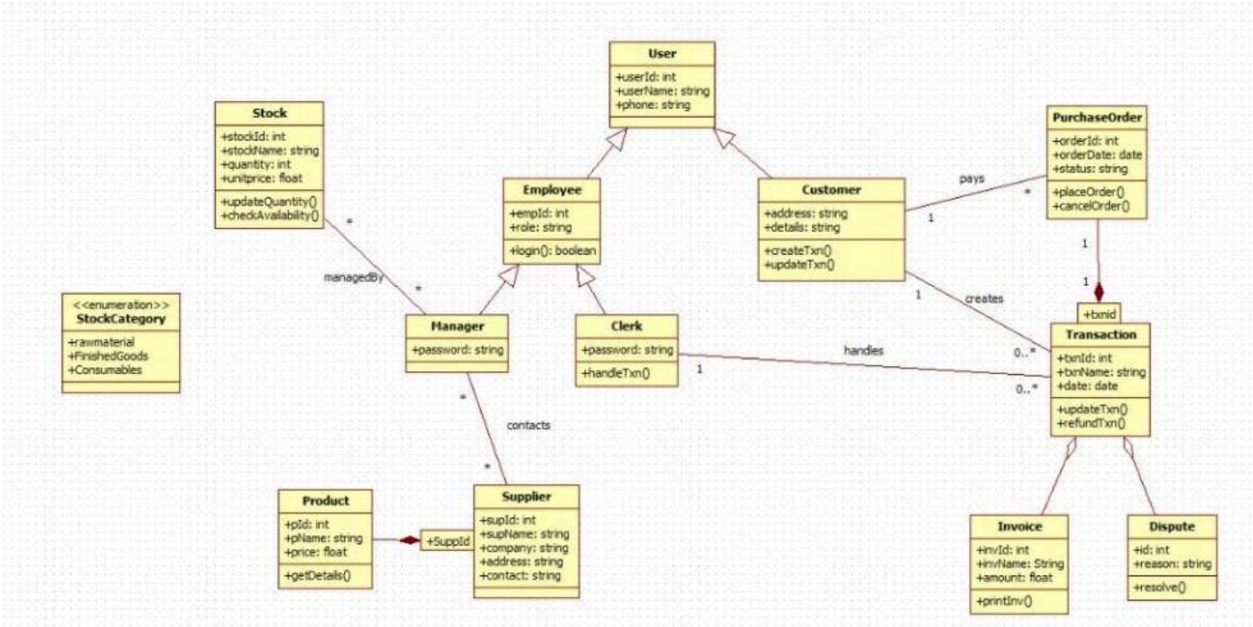

Fig 4.1 Class diagram of Stock management system

The class diagram represents an inventory and transaction management system involving users, employees, customers, suppliers, products, and stock. Customers create purchase orders, which generate transactions handled by clerks. Stock items belong to specific categories and are managed by managers who update quantity and availability. Suppliers provide products linked to stock, while invoices and disputes are associated with transactions for payment and issue resolution. Overall, the diagram shows how different entities interact to support purchasing, stock management, and transaction processing.

State diagram:

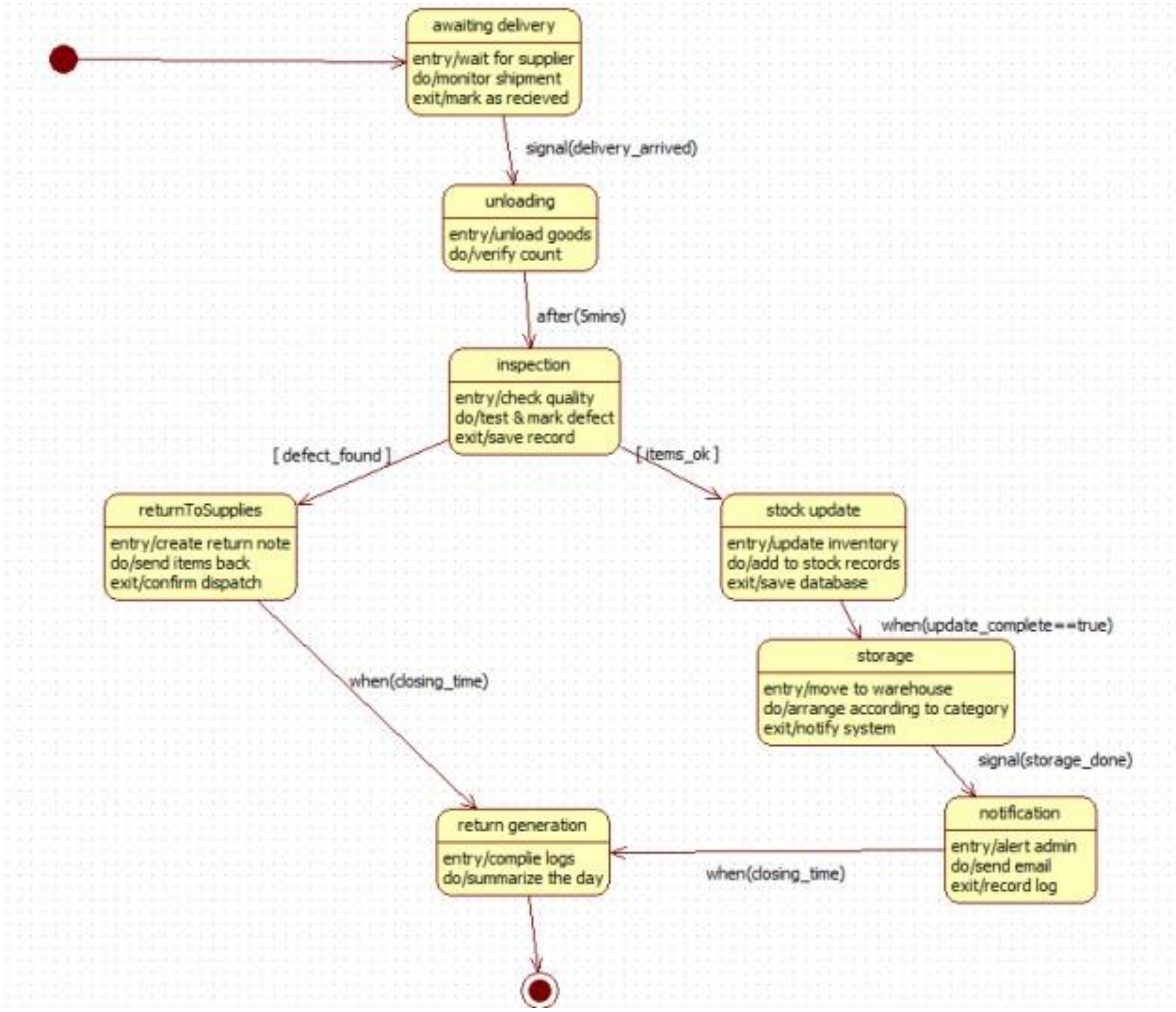

Fig 4.2 State diagram of Stock management system

The state diagram models the workflow of handling incoming goods in an inventory system. The process starts with awaiting delivery, followed by unloading and inspecting the goods. If defects are found, items are returned to the supplier; if the goods are acceptable, the system updates stock records and moves items into storage. After storage, a notification is sent to the administrator. At closing time, the system generates a daily return log, summarizing all activities before ending the process.

Use-case diagram:

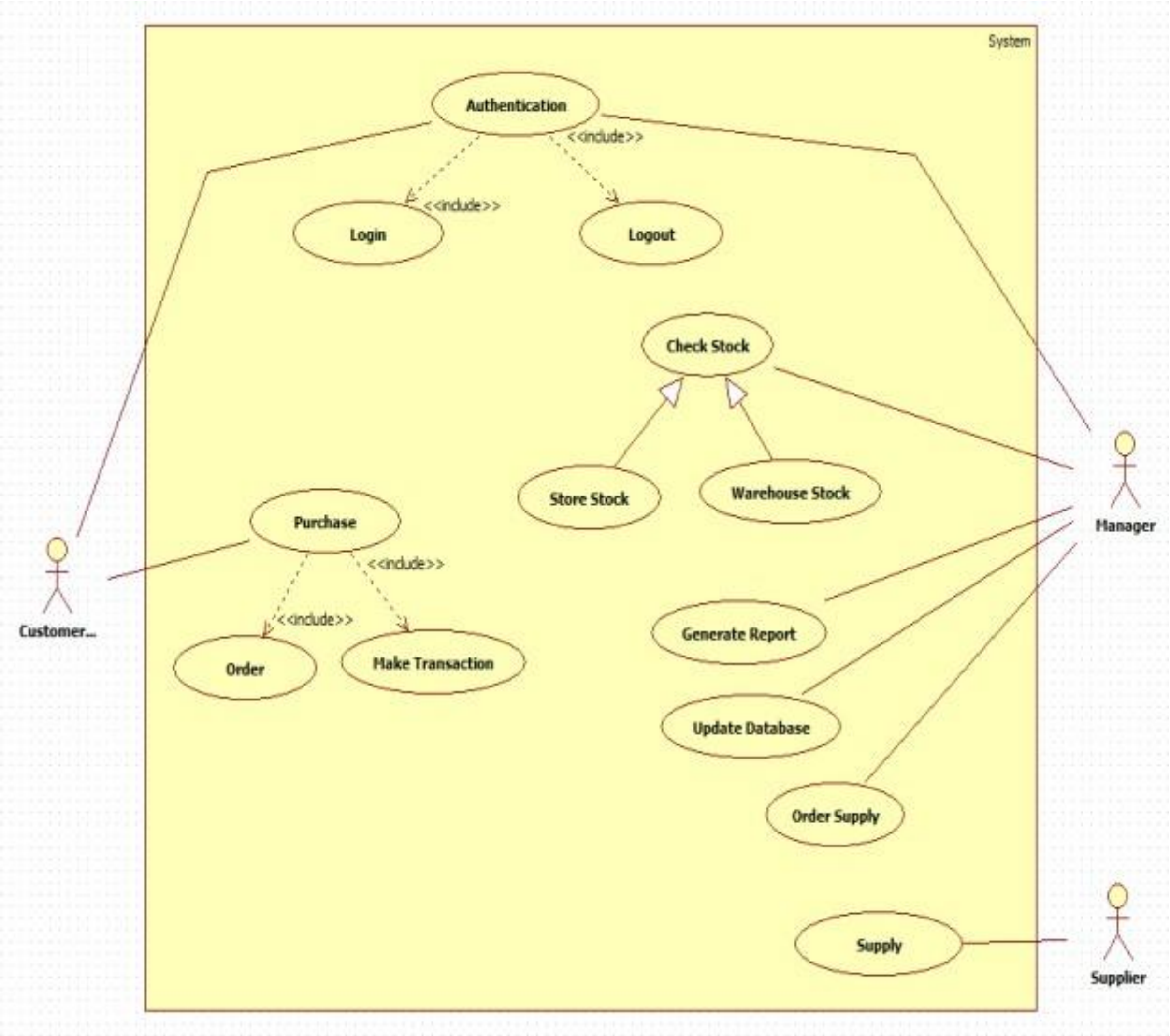

Fig 4.3 Use-case diagram of Stock management system

The use case diagram illustrates an inventory and order management system involving three actors: Customer, Manager, and Supplier. Customers can authenticate, log in, make purchases, place orders, and complete transactions. Managers oversee stock by checking warehouse and store quantities, generating reports, updating the database, and ordering supplies. Suppliers provide stock based on the manager’s supply orders. Authentication supports all major activities, ensuring secure access to system functions.

Sequence diagram:

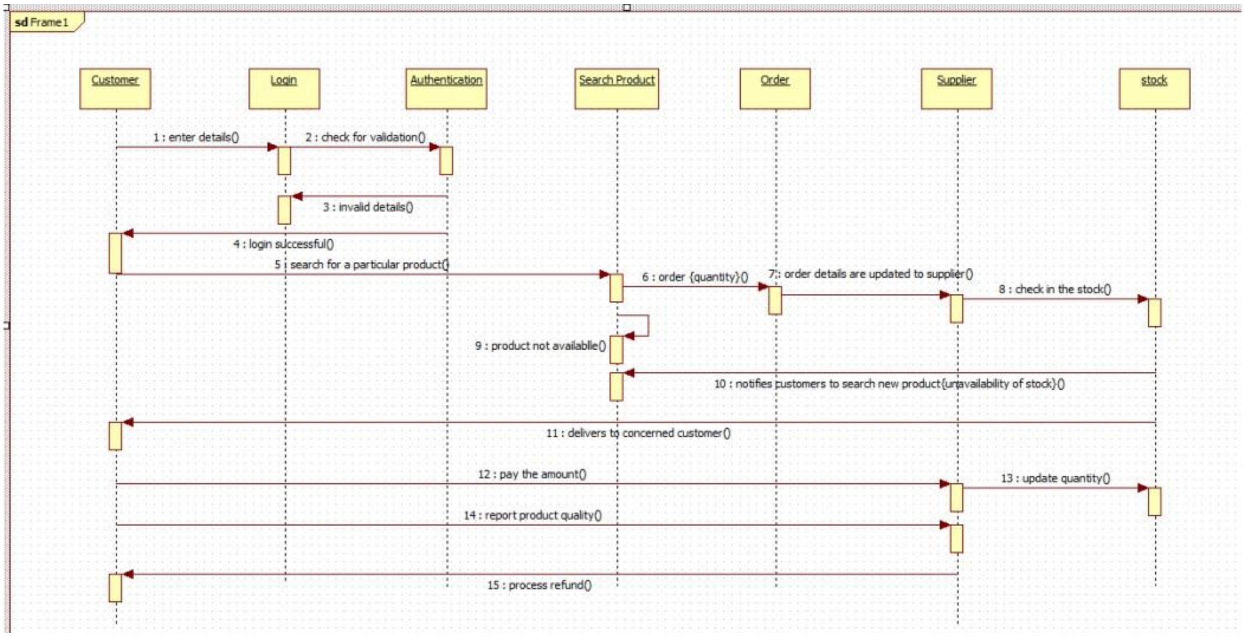
sd Frame1
Customer
Login
Authentication
Search Product
Order
Supplier
stock
1 : enter details()
2 : check for validation()
3 : invalid details()
4 : login successful()
5 : search for a particular product()
6 : order {quantity}()
7 : order details are updated to supplier()
8 : check in the stock()
9 : product not availablle()
10 : notifies customers to search new product{unavailability of stock}()
11 : delivers to concerned customer()
12 : pay the amount()
13 : update quantity()
14 : report product quality()
15 : process refund()

Fig 4.4 Sequence diagram of Stock management system

The sequence diagram illustrates the workflow of an online product ordering system. A customer logs in, searches for a product, and places an order. The system forwards order details to the supplier, who checks stock availability. If the product is unavailable, the system notifies the customer. If available, the supplier delivers the product, the customer makes the payment, and stock levels are updated. The customer may also report product quality, after which the system processes a refund if required.

Activity diagram:

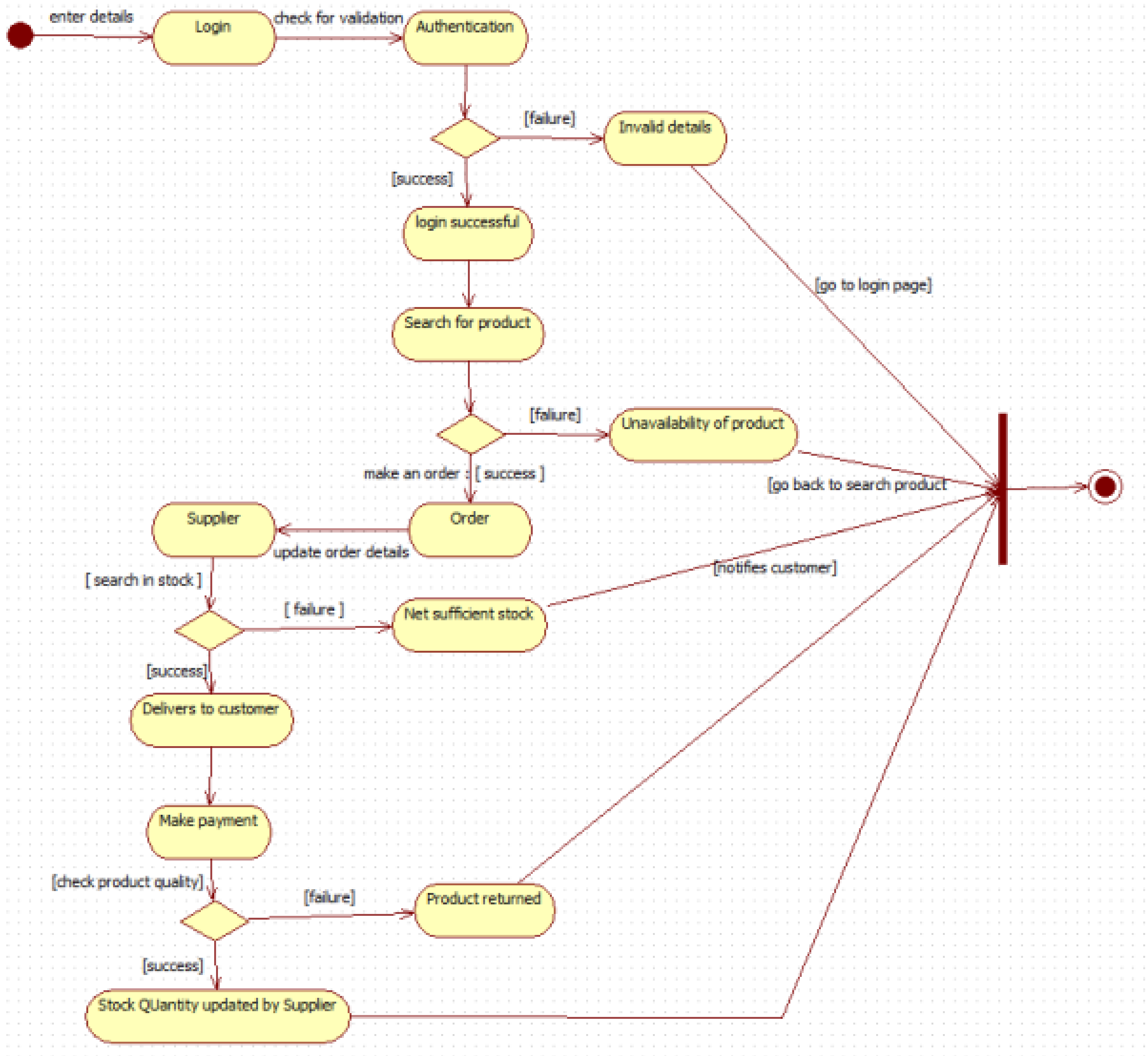

Fig 4.5 Activity diagram of Stock management system

The activity diagram shows the workflow of an online product purchase process. The customer logs in and is authenticated; if valid, they search for a product and place an order. The supplier checks stock availability and updates order details. If stock is sufficient, the product is delivered, payment is made, and product quality is checked. If the product fails the quality check, it is returned; if successful, stock quantity is updated. Invalid login or unavailable products redirect the user back to the appropriate steps.

5. Passport Automation System

Problem Statement

Traditional passport application and verification processes involve multiple manual steps, long queues, and paperwork, leading to delays, errors, and inefficiencies. Tracking application status, verifying documents, scheduling appointments, and issuing passports become increasingly difficult as the number of applicants grows.

A Passport Automation System is needed to streamline application submission, document verification, appointment scheduling, status tracking, and passport issuance. The goal is to reduce processing time, minimize human errors, improve transparency, and provide a faster, user-friendly experience for both applicants and officials through a centralized digital platform.

Software Requirements Specification (SRS)

1. Introduction

The Passport Automation System aims to automate the entire passport application process, allowing applicants to complete their submissions online and making verification faster for authorities. It replaces time-consuming manual workflows with a digital process that minimizes paperwork and reduces errors. The system provides applicants with clear guidance, submission tracking, and timely updates. Its goal is to make passport processing more transparent, efficient, and user-friendly for both applicants and officials.

2. Overall Description

The system manages applicant registration, document uploads, appointment scheduling, fee payments, police verification, and application tracking. Separate interfaces are available for applicants, passport officers, and police departments. A centralized database stores all documents, application details, and verification reports. The system provides automated notifications for each stage of the process, such as appointment confirmations and application approvals. It ensures that the entire process is streamlined and monitored with minimal manual intervention.

3. External Interface Requirements

The system includes a web portal for applicants and a secure dashboard for passport officials and police staff. It connects to a central database to store documents and application details. It may send SMS or email alerts for updates such as appointment reminders, document issues, and verification results. Payment gateways may be integrated to handle application fees. Printers can be used for generating physical receipts or acknowledgment letters.

4. System Features (Functional Requirements)

- FR1: The system shall allow applicants to create accounts and fill application forms online.
- FR2: The system shall allow users to upload scanned documents such as ID proof and address proof.
- FR3: The system shall validate mandatory fields and document formats.

- FR4: The system shall provide available appointment slots and allow users to book one.
- FR5: The system shall update application status after document checking, biometrics, and police verification.
- FR6: The system shall store police verification reports in the applicant's record.
- FR7: The system shall allow passport officers to approve, reject, or mark applications for re-verification.
- FR8: The system shall display live status updates to applicants at every stage.
- FR9: The system shall record fee payments and generate digital receipts.
- FR10: The system shall archive completed applications for future reference.
- FR11: The system shall notify applicants of missing documents or errors.
- FR12: The system shall allow administrators to generate statistical reports (daily/weekly/monthly).

5. Non-Functional Requirements

- NFR1: The system shall ensure strong security for sensitive personal information.
- NFR2: It shall provide fast and smooth navigation, even during high-traffic periods.
- NFR3: The system shall remain operational 24/7 for online submissions and tracking.
- NFR4: It shall comply with government security standards and identity verification protocols.
- NFR5: The interface shall be simple and accessible to users with minimal technical knowledge.
- NFR6: Document uploads shall be processed and stored efficiently without corruption.
- NFR7: The system shall support automated backups to prevent data loss.
- NFR8: It shall ensure consistent and accurate synchronization across all verification departments.
- NFR9: The system shall support scalability for large numbers of users and applications.
- NFR10: The system shall maintain complete audit logs for legal and administrative purposes.

Class diagram:

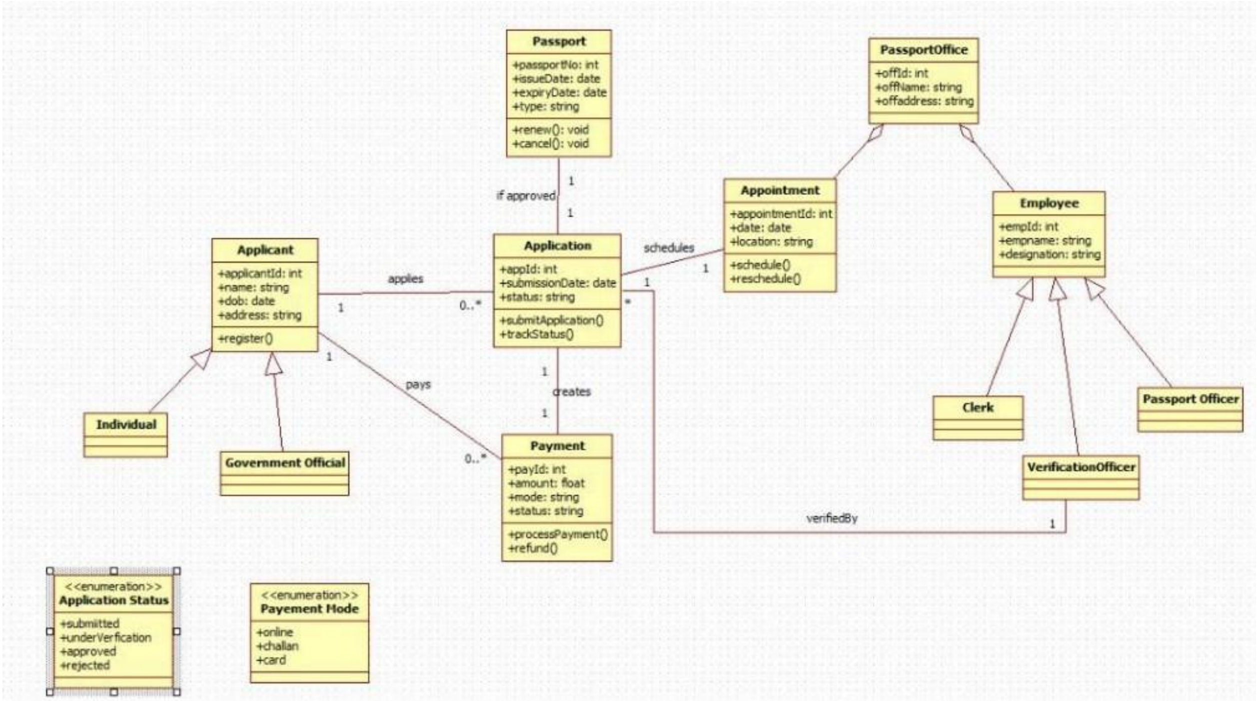

Fig 5.1 Class diagram of Passport Automation System

The class diagram represents a passport application management system. An Applicant—either an individual or government official—submits an Application and makes the required Payment. The Passport Office manages appointments and employs different staff roles, including clerks, passport officers, and verification officers, who verify applications. Upon approval, a Passport is issued or renewed. The diagram also includes application statuses and payment modes, showing how the system processes applications from submission to approval.

State diagram:

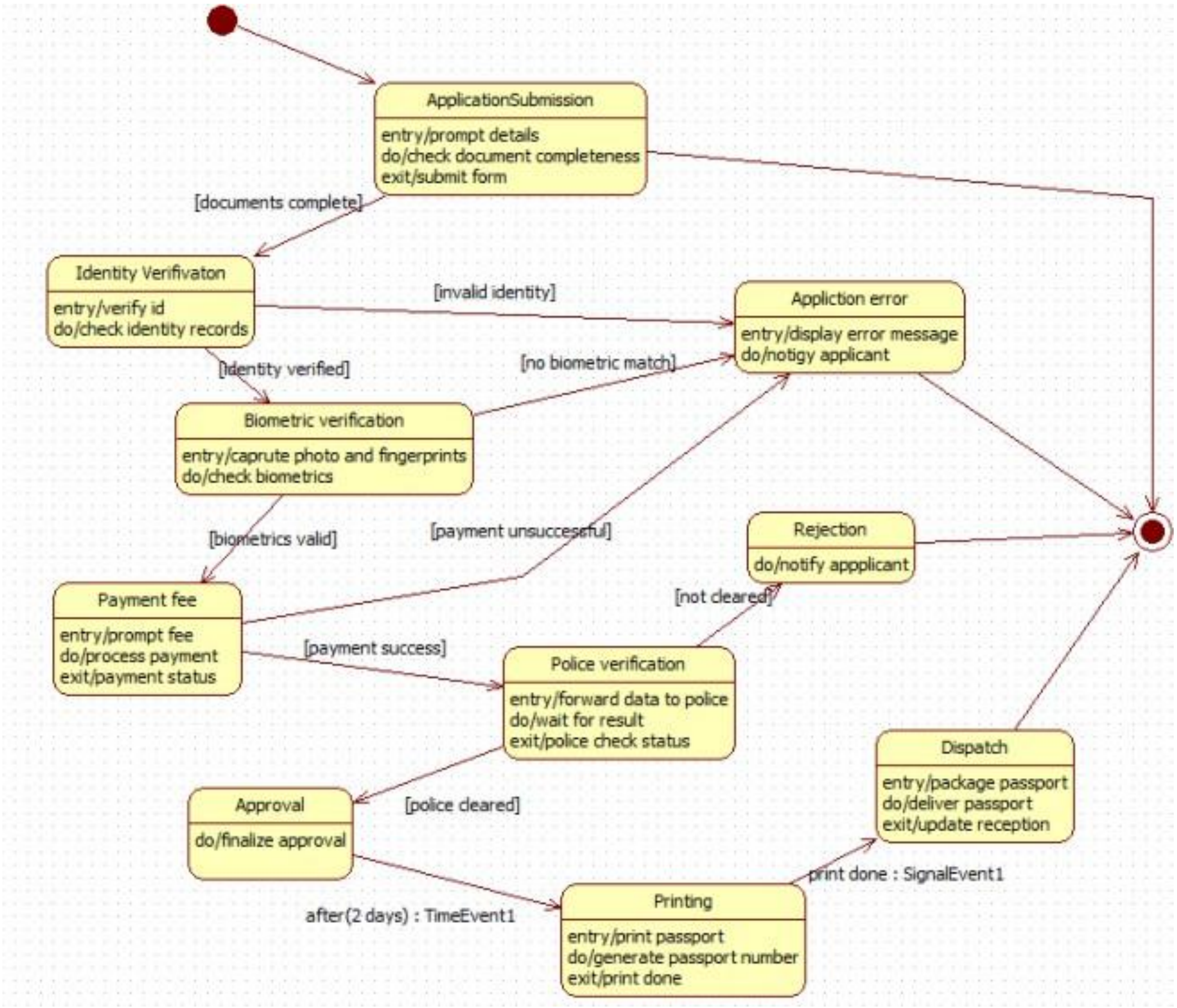

Fig 5.2 State diagram of Passport Automation System

The state diagram illustrates the complete workflow of a passport application process. It begins with application submission, followed by identity and biometric verification. If verification fails, the system moves to an error or rejection state. Upon successful verification, the applicant pays the required fee, and the application proceeds to police verification. Once cleared, the passport is approved, printed, and dispatched to the applicant. Any failure at intermediate steps leads to rejection.

Use-case diagram:

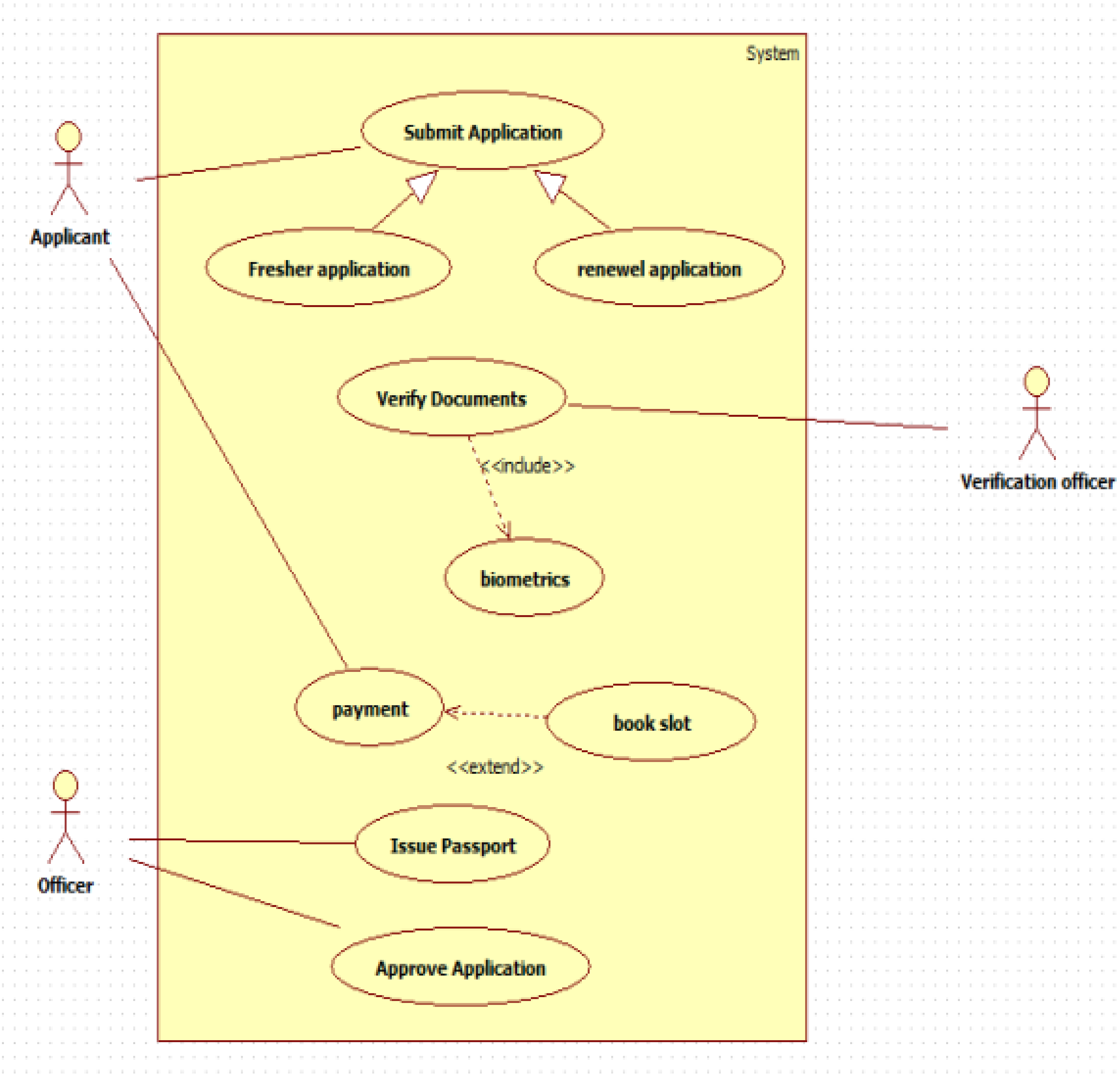

Fig 5.3 Use-case diagram of Passport Automation System

The use case diagram represents a passport application system involving Applicants, Officers, and Verification Officers. Applicants submit either a fresh or renewal application, after which the system verifies documents and captures biometrics. Applicants then make payments and may book a slot if required. Officers handle the approval and issuance of the passport, completing the application process.

Sequence diagram:

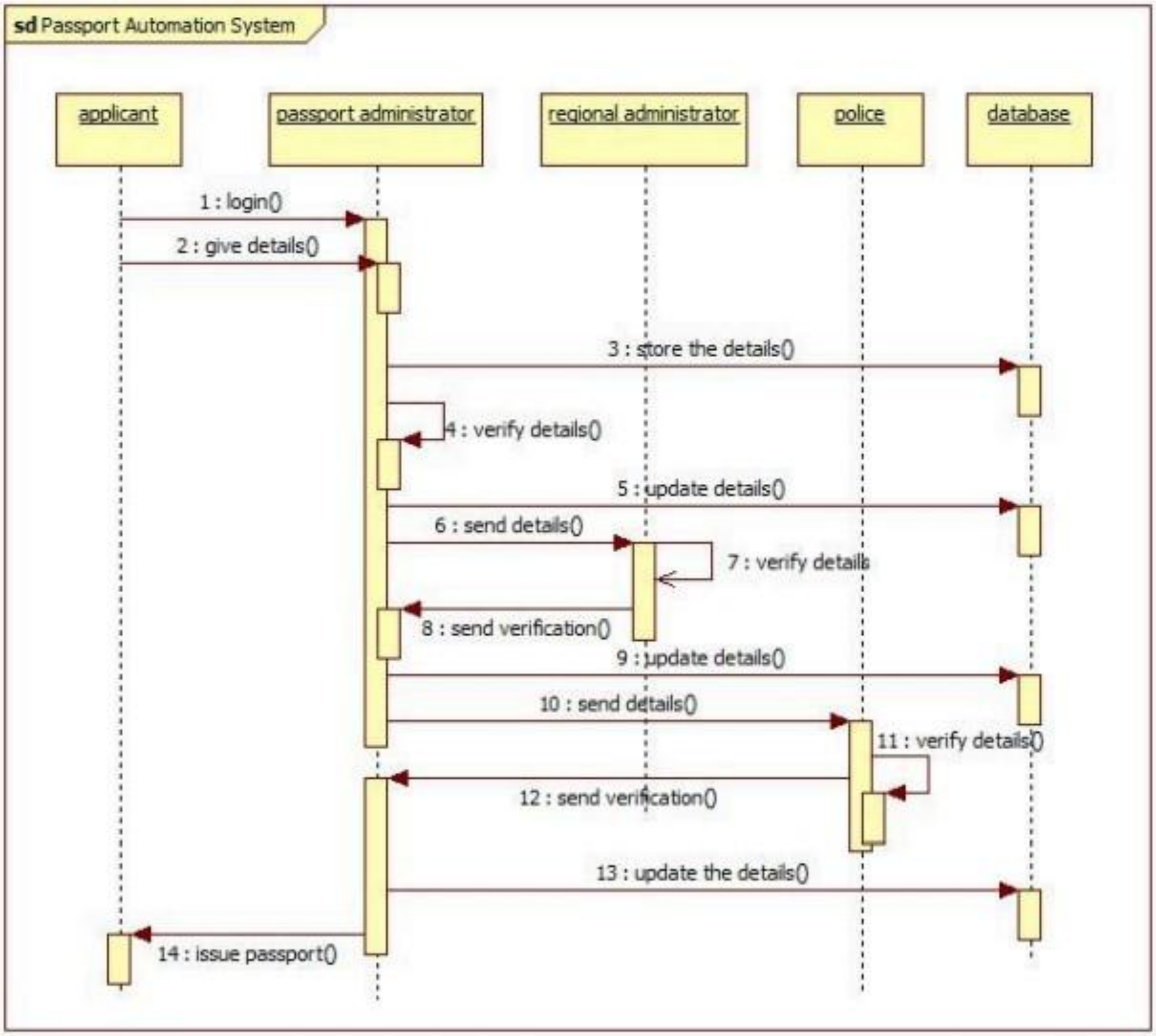

Fig 5.4 Sequence diagram of Passport Automation System

The sequence diagram shows the workflow of a passport automation system. The applicant logs in and submits details, which the passport administrator stores in the database. The regional administrator and police department sequentially verify the submitted information and update the records. After receiving verification from both authorities, the passport administrator issues the passport to the applicant.

Activity diagram:

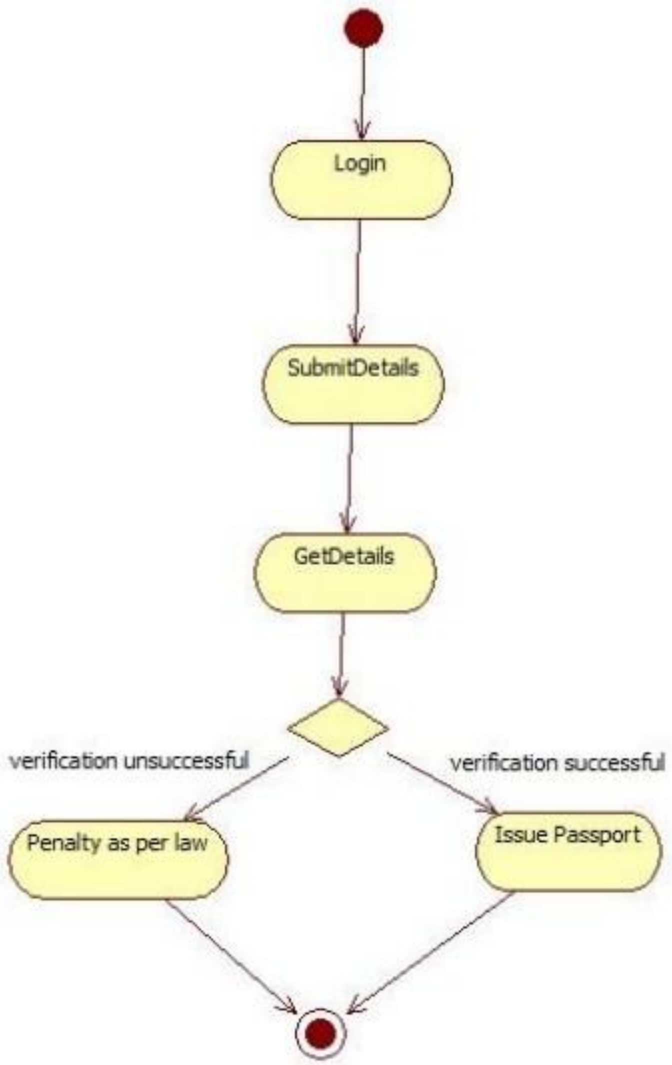

Fig 5.5 Activity diagram of Passport Automation System

The activity diagram shows a simple passport verification process. A user logs in, submits their details, and the system retrieves and checks the information. If verification is successful, a passport is issued; if verification fails, a legal penalty is applied. The process then ends.