MIT ART DESIGN & TECHNOLOGY UNIVERSITY

MIT College of Management (MITCOM), Pune

PROGRAMME: MASTER OF COMPUTER APPLICATION (MCA CC/DS)

CERTIFICATE

This is to certify that, Mr./Miss _____has submitted a Practical Report on _____ to MIT – ADT University, Pune for the partial fulfillment of Master in Computer Application (Data Science/ Cloud Computing) submitted during the academic year 2024-25.

PRN No._____MCA Year_____MCA Sem. – _____

Subject Incharge	Dr.Alkawati Magadum HOD MCA	Dr.Sangita Phunde Principal

External Examiner **Sign of Examiners:**

1._____ _____

Internal Examiner **Sign of Examiners:**

2._____ _____

MIT ART DESIGN & TECHNOLOGY UNIVERSITY

MIT College of Management (MITCOM), Pune

Declaration

I undersigned hereby declares that, the Journal of assignments solved by me and it is executed as per the course requirement of MCA program of MIT-ADT University, Pune. This report has not submitted by me or any other person to any other University or Institution for a degree or diploma course. This is my own and original work.

Place: MITCOM, Pune
Date:

Sign of the student: ----------

Name of the Student.....

MIT ART DESIGN & TECHNOLOGY UNIVERSITY

MIT College of Management (MITCOM), Pune

Sub:-

Name:- Div:- MCA

Sr No.	Name Of The Practical	Page	Date	Record Sign
1	**Restaurant management system**			
2	**ATM**			
3	**Online Shopping website**			
4	**Hospital management system**			
5	**Library Management System**			

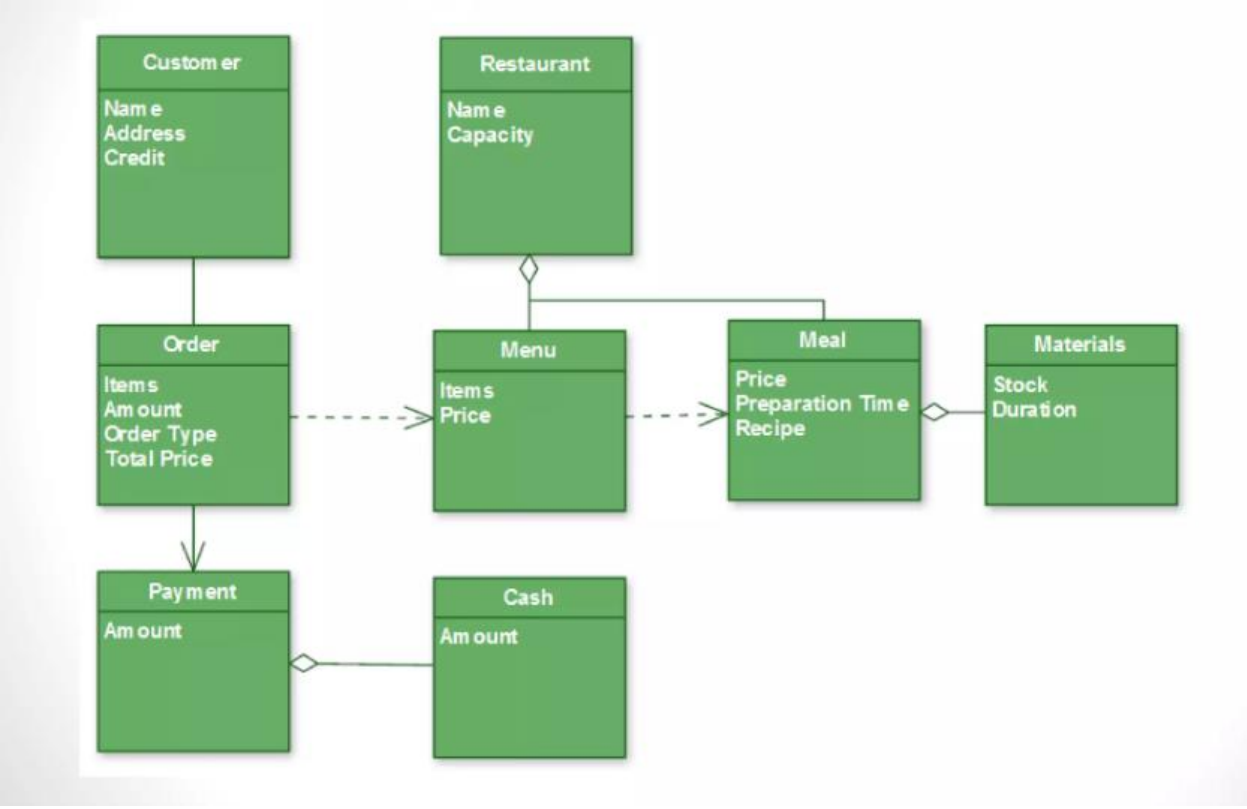
UML: **Restaurant management system**

Restaurant Services
UML is a tool to design services with a unified language. A simple restaurant case is chosen to be described by UML. We show the basic processes of a restaurant including:

- Customer entry to the restaurant
- To provide Menu and take Order
- To Order food in order to Cook
- To Cook the ordered food
- Serving food to customer
- Payment activities

Class Diagram, Use Case Diagram, Sequence Diagram and State Machine Diagram are used to show a basic process in a restaurant.

Class Diagram



is a static structure diagram in UML (Unified Modeling Language) that depicts the classes, their attributes, and their relationships.
Here's a breakdown of the classes and relationships shown in the diagram:
Classes:

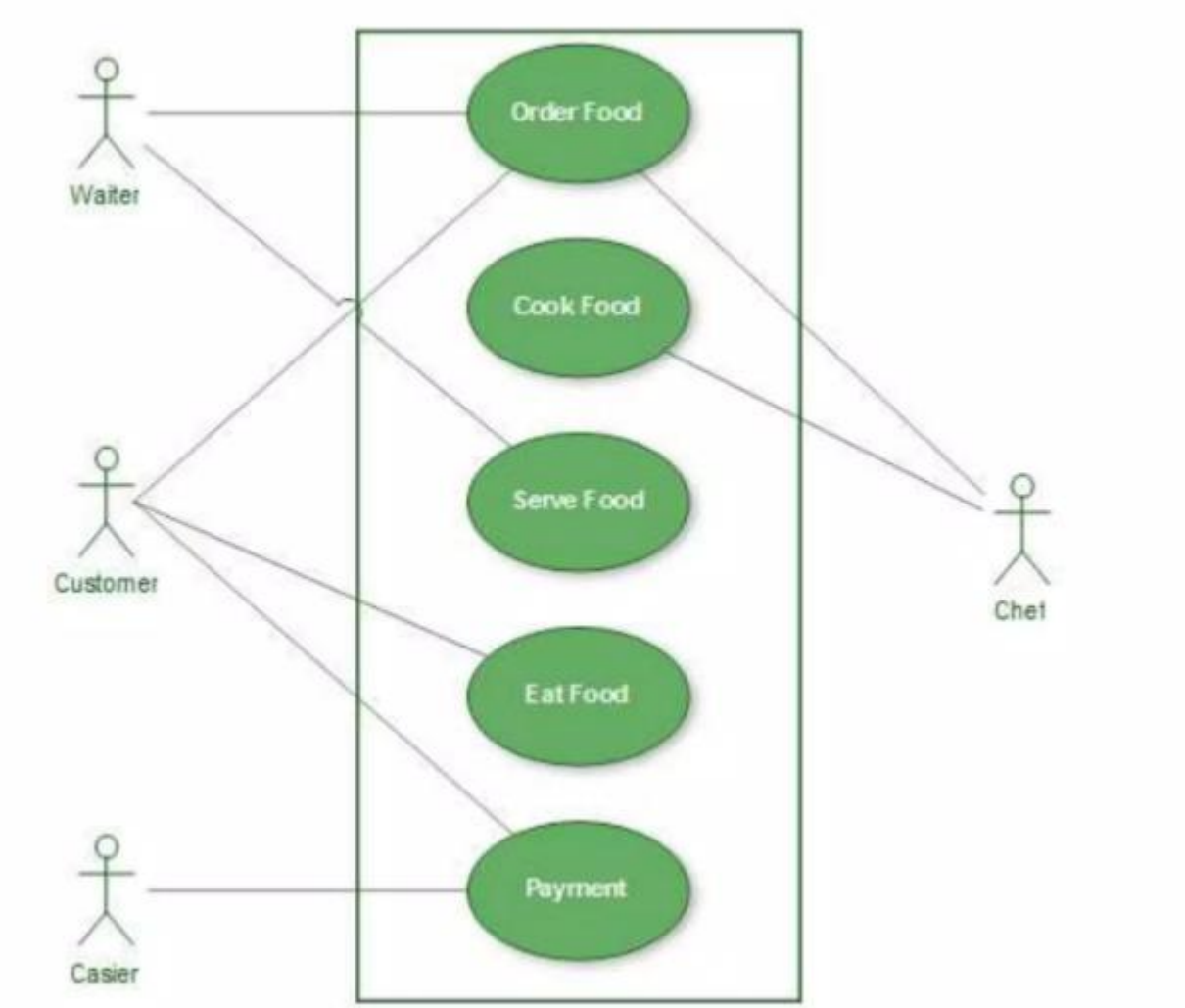
- **Customer:** Represents a customer with attributes like Name, Address, and Credit.
- **Restaurant:** Represents a restaurant with attributes like Name, Address, and Capacity.
- **Order:** Represents an order placed by a customer with attributes like Items, Amount, Order Type, and Total Price.
- **Menu:** Represents a menu with attributes like Items and Price.
- **Meal:** Represents a meal with attributes like Price, Preparation Time, and Recipe.
- **Materials:** Represents materials used in preparing meals with attributes like Stock and Duration.
- **Payment:** Represents a payment made by a customer with the attribute Amount.
- **Cash:** Represents a cash payment with the attribute Amount.

Relationships:

- **Customer - Order:** A customer can place multiple orders, represented by a one-to-many relationship.
- **Restaurant - Order:** A restaurant can process multiple orders, represented by a one-to-many relationship.
- **Order - Menu:** An order can contain multiple menu items, represented by a many-to-many relationship.
- **Order - Payment:** An order can have multiple payments (e.g., cash, credit card), represented by a one-to-many relationship.
- **Meal - Materials:** A meal requires multiple materials, represented by a many-to-many relationship.

Overall, the diagram represents a simplified model of a restaurant system, showing the entities involved and their relationships. It provides a visual representation of the system's structure and can be used as a basis for further development and analysis.

Use Case Diagram



This type of diagram is used in software engineering to visualize the interactions between users (actors) and the system.
Here's a breakdown of the elements in your diagram:

Actors:

- **Waiter:** Represents the restaurant staff taking orders.
- **Customer:** Represents the person placing the order.
- **Chef:** Represents the person cooking the food.
- **Cashier:** Represents the person taking payment.

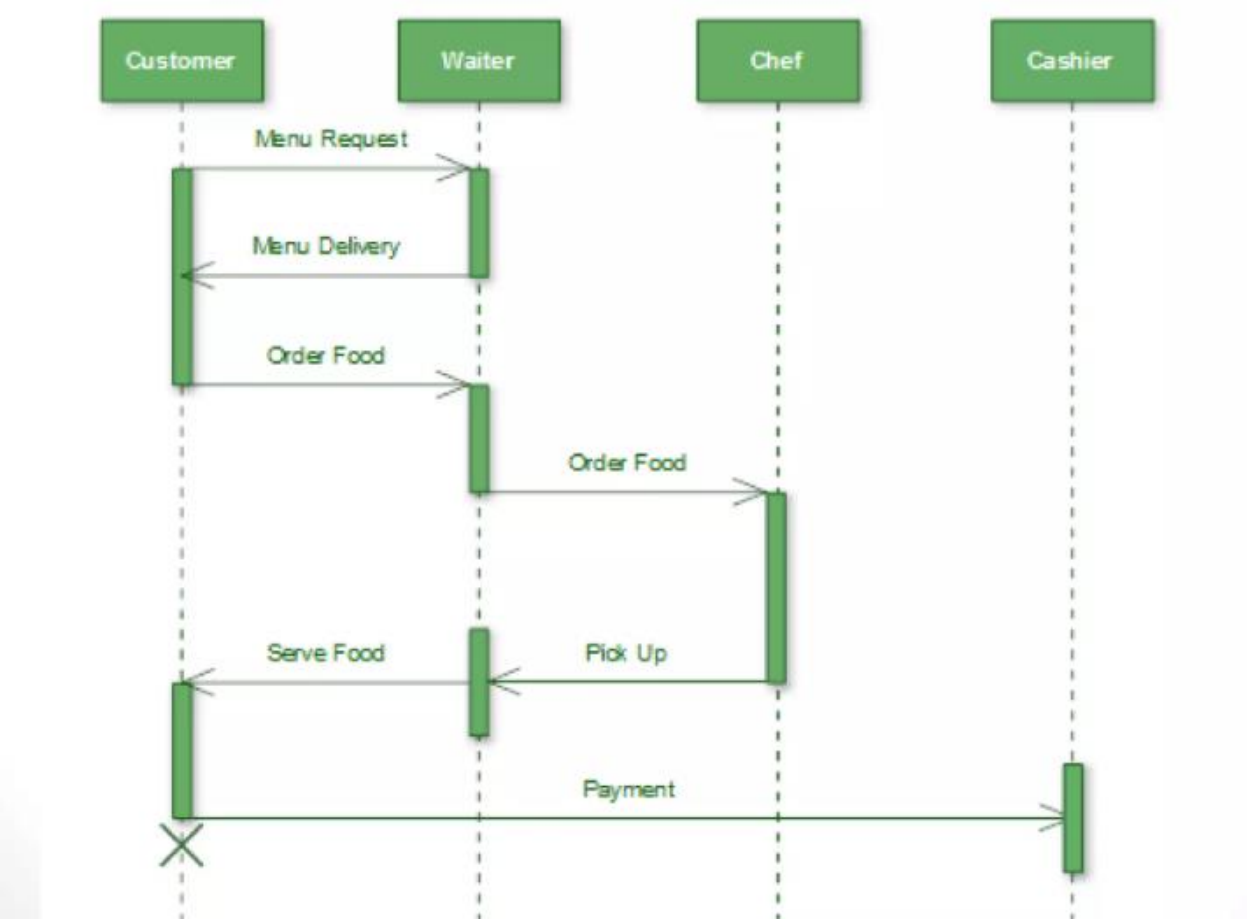
Use Cases:

- **Order Food:** This use case involves the waiter taking the customer's order.
- **Cook Food:** This use case involves the chef preparing the food based on the order.

- **Serve Food:** This use case involves the waiter serving the food to the customer.
- **Eat Food:** This use case involves the customer consuming the food.
- **Payment:** This use case involves the customer paying the cashier.

Relationships:
The lines connecting the actors to the use cases represent the interactions between them. For example, the customer interacts with the waiter to order food, and the waiter interacts with the chef to cook the food.

Sequence Diagram



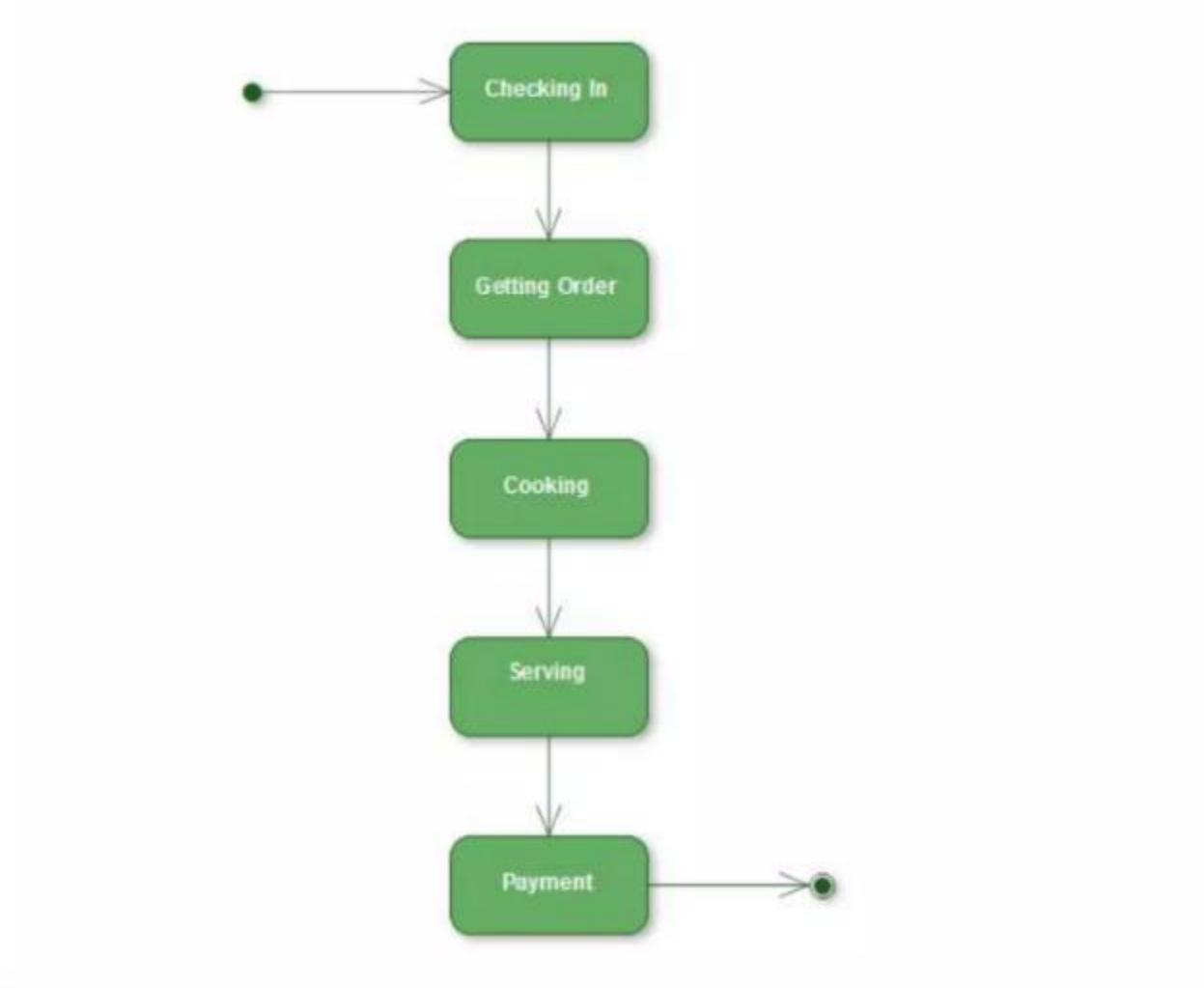
This sequence diagram illustrates the interactions between different actors in a restaurant scenario. Let's break down the sequence of events:

1. **Customer Requests Menu:**
    - The customer sends a "Menu Request" message to the waiter.
2. **Waiter Delivers Menu:**
    - The waiter receives the request and sends the "Menu Delivery" message back to the customer.
3. **Customer Orders Food:**
    - The customer sends an "Order Food" message to the waiter.
4. **Waiter Places Order:**

    - The waiter receives the order and sends an "Order Food" message to the chef.
5. **Chef Cooks Food:**
    - The chef receives the order and starts cooking the food.
6. **Chef Prepares Food:**
    - The chef sends a "Pick Up" message to the waiter.
7. **Waiter Serves Food:**
    - The waiter picks up the food from the chef and sends a "Serve Food" message to the customer.
8. **Customer Eats Food:**
    - The customer receives the food and eats it.
9. **Customer Pays:**
    - The customer sends a "Payment" message to the cashier.
10. **Cashier Receives Payment:**

- The cashier receives the payment and completes the transaction.

State Machine Diagram



This diagram represents a State Machine Diagram, which is used to model the behavior of a system over time. In this specific case, it depicts the different states that a customer's order goes through in a restaurant.
Here's a breakdown of the states and transitions:
States:

- **Checking In:** The initial state when a customer enters the restaurant.
- **Getting Order:** The state where the waiter is taking the customer's order.
- **Cooking:** The state where the kitchen is preparing the food.
- **Serving:** The state where the waiter is serving the food to the customer.
- **Payment:** The final state where the customer is paying for the meal.

Transitions:

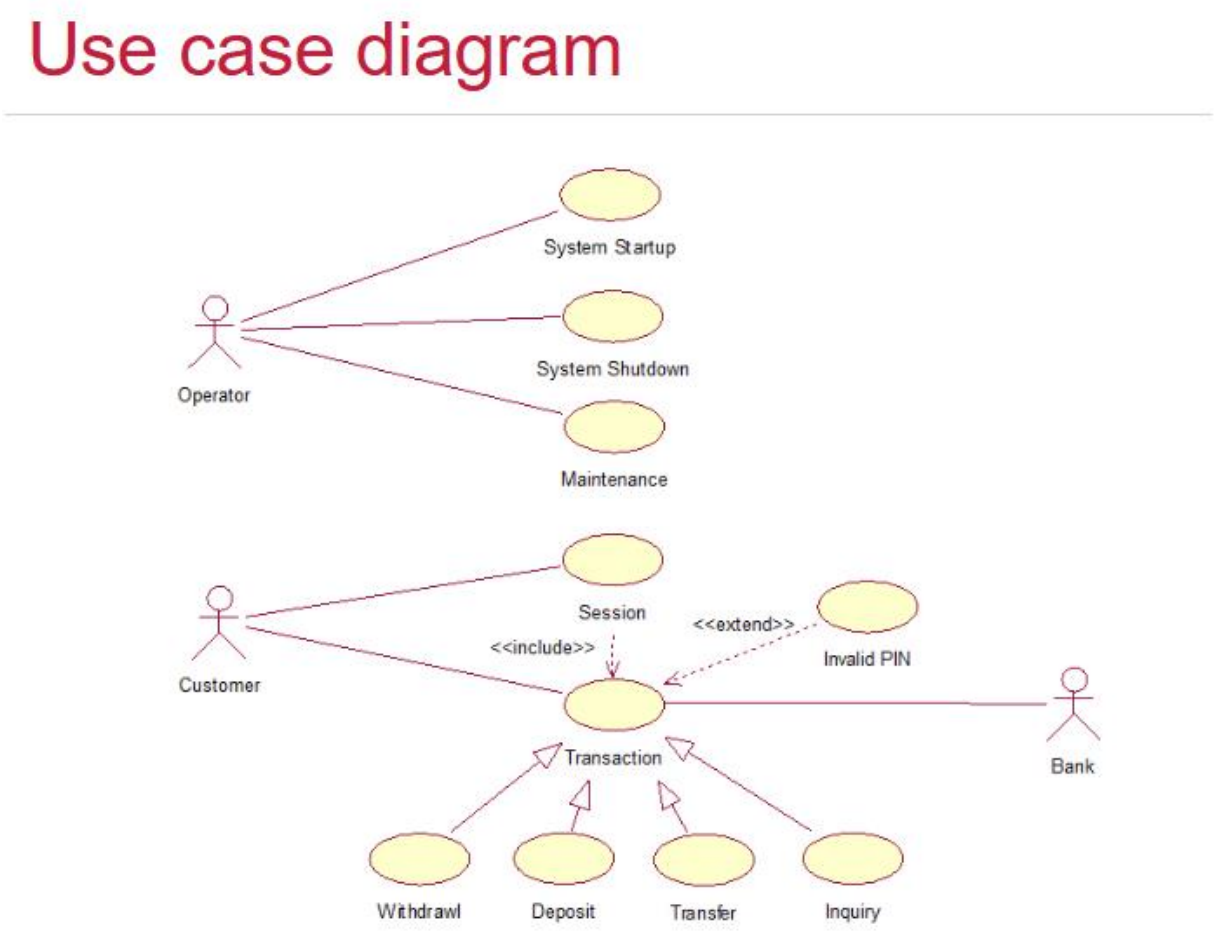
- **Checking In -> Getting Order:** The customer is seated and ready to order.

- **Getting Order -> Cooking:** The waiter places the order with the kitchen.
- **Cooking -> Serving:** The food is prepared and ready to be served.
- **Serving -> Payment:** The customer finishes their meal and is ready to pay.

This state machine diagram provides a clear visual representation of the different stages involved in the process of dining in a restaurant. It helps to understand the flow of the process and identify potential bottlenecks or areas for improvement.

ATM

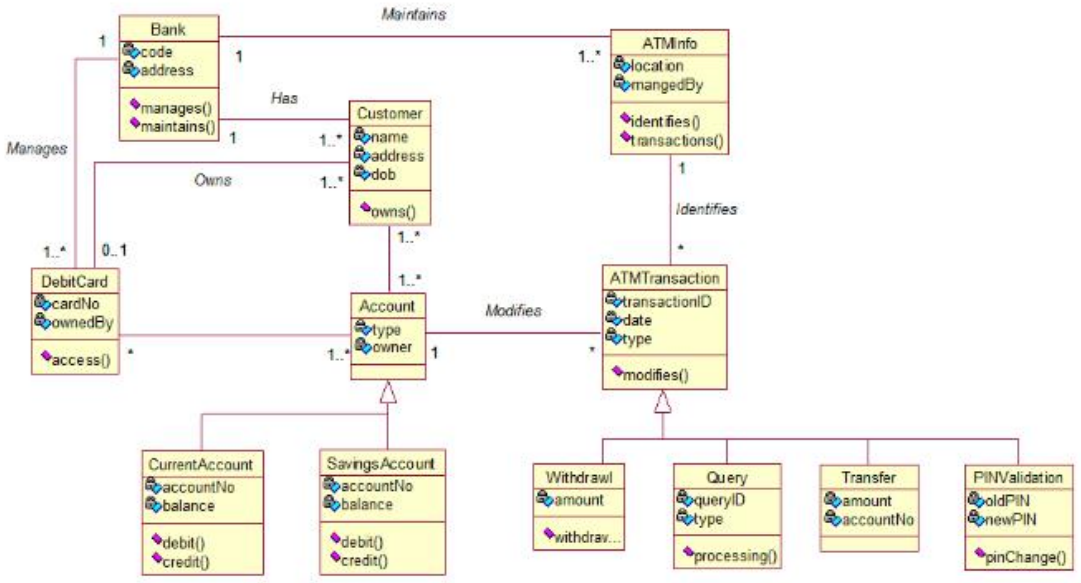
For an **ATM** system, we can use similar UML diagrams to model the basic processes. Here, we will describe key activities involved in the ATM system, such as card insertion, PIN verification, money withdrawal, and printing receipts. We will break it down with the same UML diagrams: Class Diagram, Use Case Diagram, Sequence Diagram, and State Machine Diagram.



1. Use Case Diagram for ATM
The **Use Case Diagram** will represent the high-level interactions between the user (Customer) and the ATM system. The main actors are the **Customer** and the **ATM System**, and the use cases are the core operations the ATM provides.
Use Cases:

- Insert Card
- Enter PIN
- Select Transaction Type (e.g., Withdrawal, Balance Inquiry, Fund Transfer)
- Withdraw Cash
- Print Receipt
- Check Balance
- End Session
- Transfer Funds

Class diagram

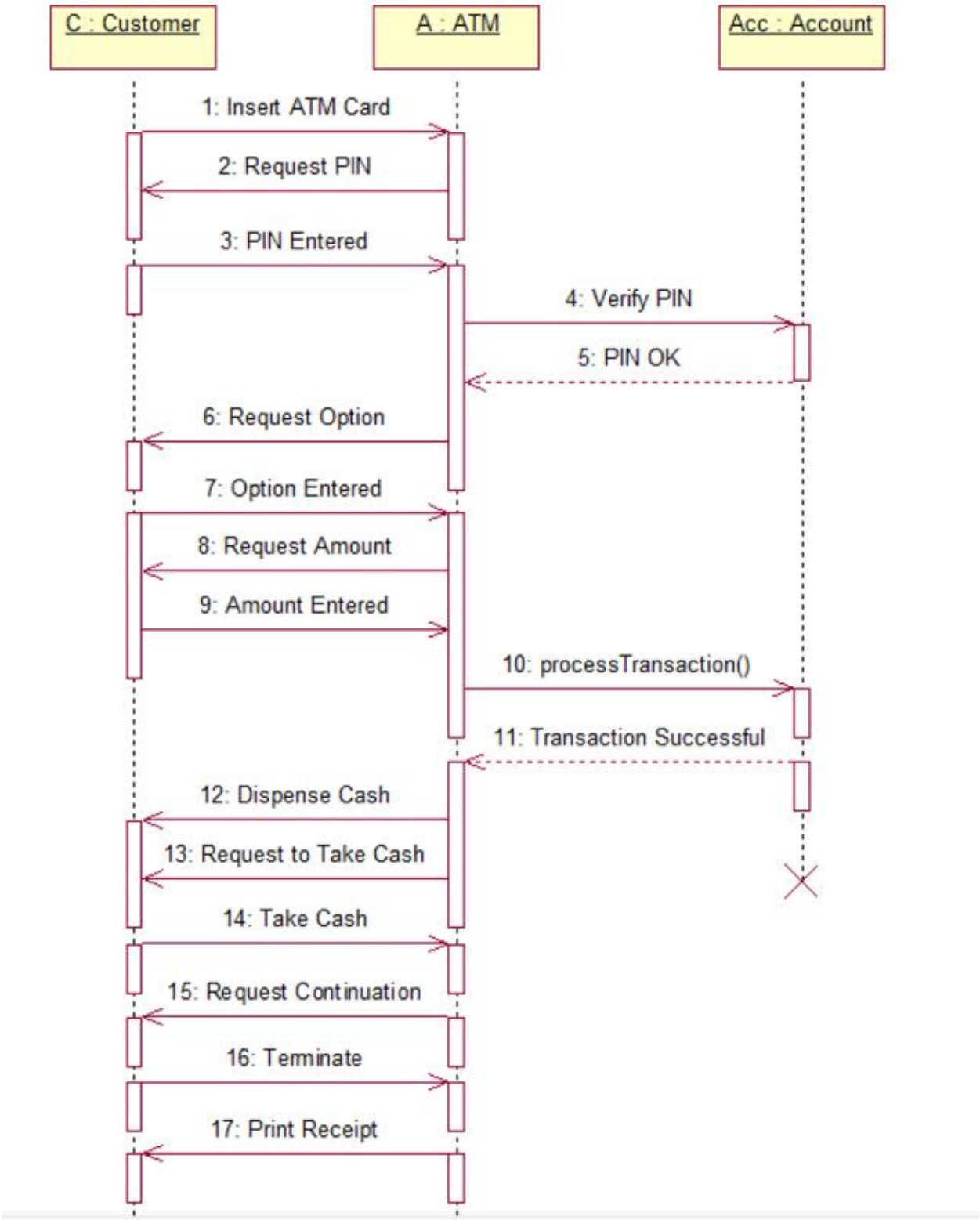


Class Diagram for ATM
The **Class Diagram** will show the structure of the ATM system, including key classes and their relationships.
Classes:

- **ATM**: The main class representing the ATM machine.

- **Card**: Represents the ATM card inserted by the customer.
- **Transaction**: A general class for different types of transactions (e.g., Withdrawal, Balance Inquiry).
- **CashDispenser**: The machine that dispenses cash during withdrawal.
- **BankAccount**: Represents the customer's bank account.

Sequence Diagram



Sequence Diagram for ATM

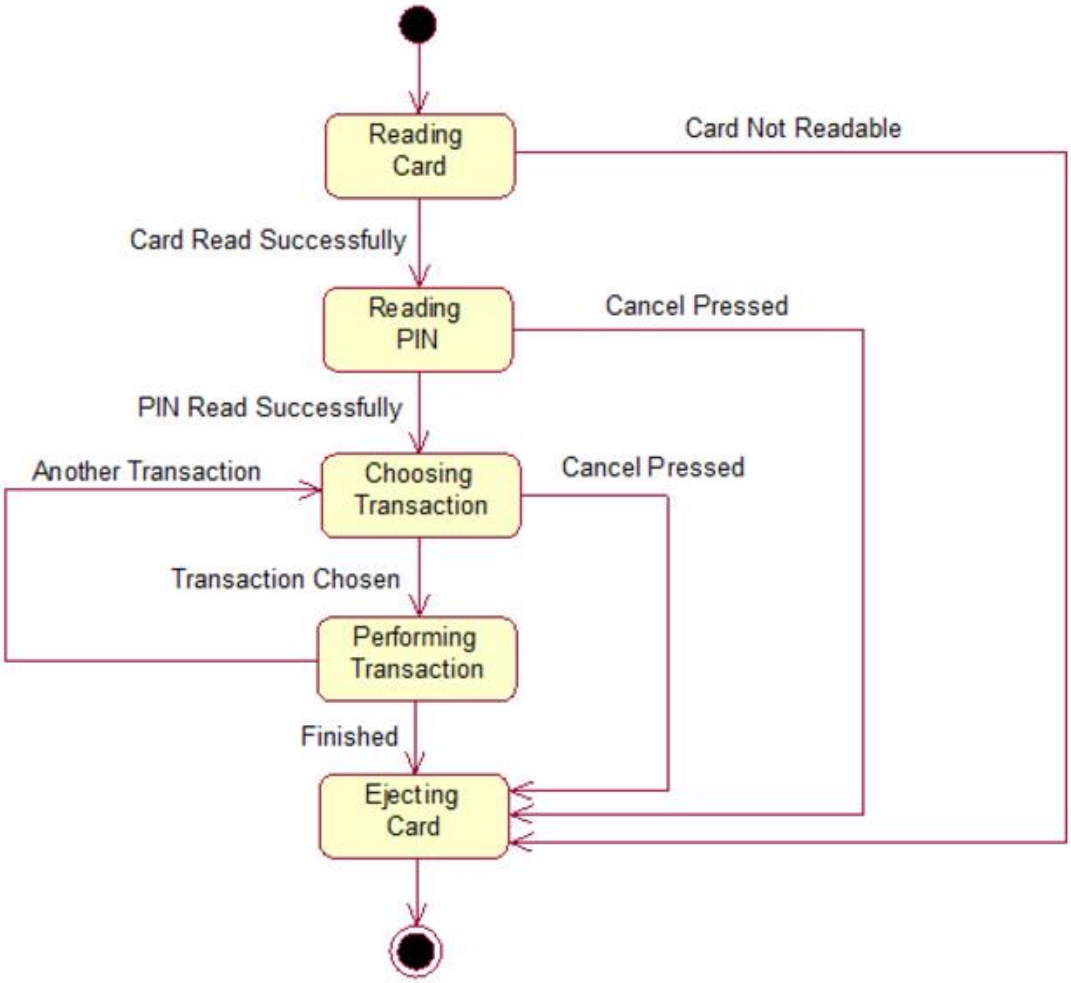
The **Sequence Diagram** will model the flow of interactions between the **Customer**, **ATM**, **BankAccount**, and other relevant objects when performing a **Withdraw Cash** operation.

Scenario: Customer withdraws money

Objects:

- **Customer**
- **ATM**
- **Card**
- **BankAccount**
- **CashDispenser**
- **ReceiptPrinter**

Statechart diagram



State Machine Diagram for ATM
The **State Machine Diagram** will describe the possible states of the ATM machine during its operation, focusing on the ATM's state transitions as the customer interacts with it.
States:

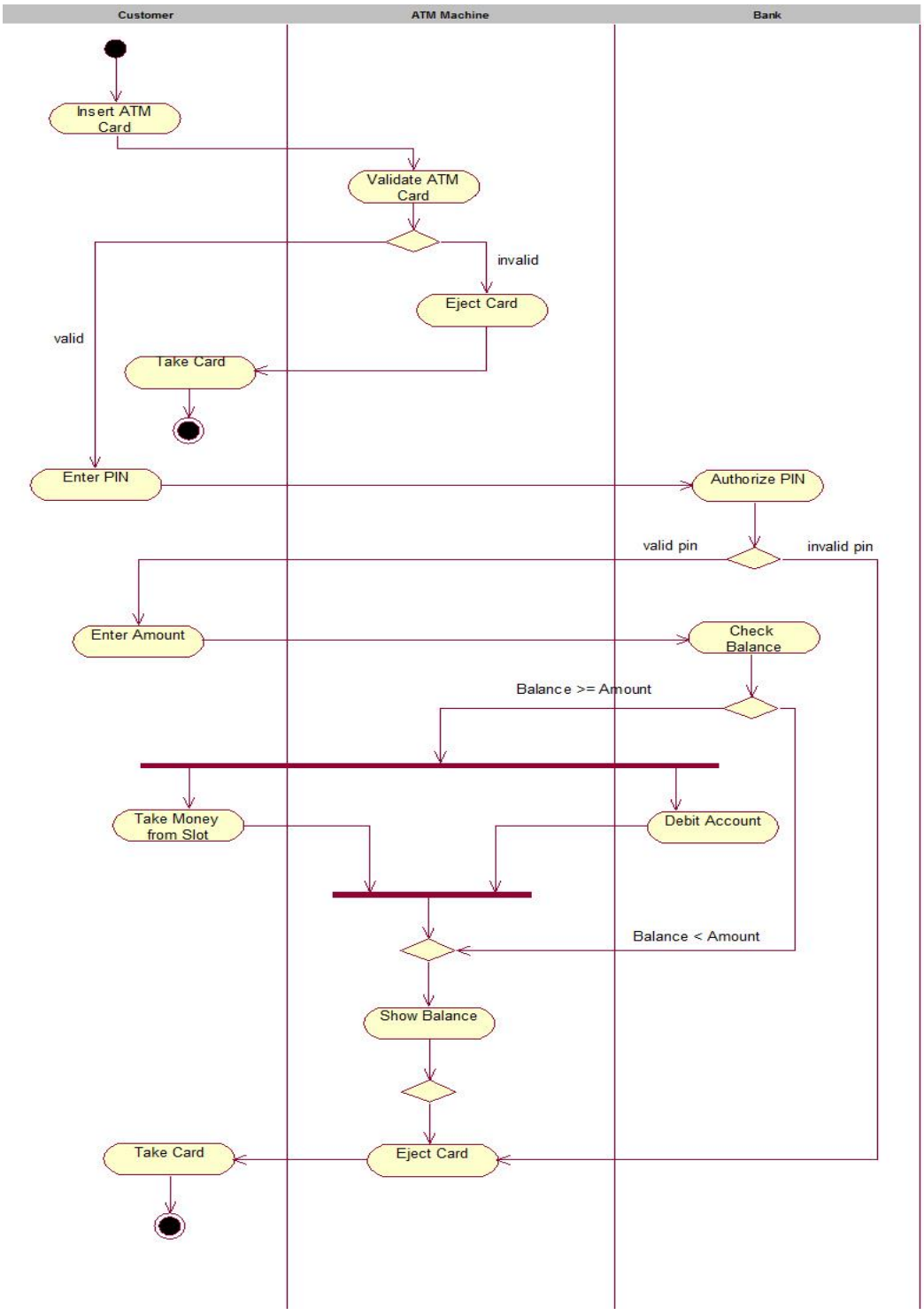
- **Idle**: The ATM is waiting for a card to be inserted.
- **Card Inserted**: The ATM waits for the user to enter a PIN.
- **PIN Entered**: The ATM verifies the PIN and checks the account.
- **Transaction Started**: The ATM processes the selected transaction (e.g., withdrawal).

- **Transaction Completed**: The transaction is finished, and the ATM is ready to print the receipt.
- **Card Ejected**: The ATM ejects the card and waits for the next user.

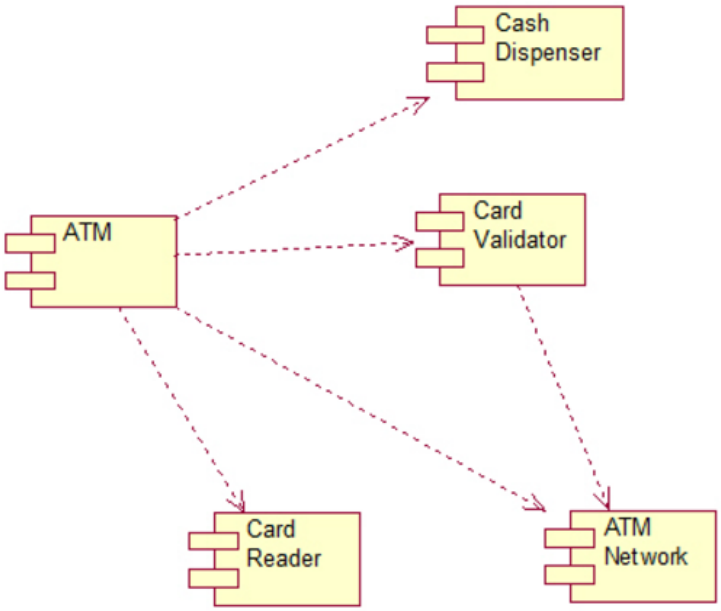
Activity diagram

ATM Activity Diagram
Key Activities:

1. Insert Card: The customer inserts the ATM card into the machine.
2. Enter PIN: The customer enters their PIN.
3. Validate PIN: The ATM verifies the entered PIN.
4. Select Transaction Type: The customer chooses the type of transaction (e.g., withdrawal, balance inquiry).
5. Perform Transaction: The ATM executes the transaction.
    - If it's a withdrawal, the system checks for sufficient balance and dispenses cash.
    - If it's a balance inquiry, it displays the balance.
6. Print Receipt: The ATM prints a receipt for the transaction.
7. Eject Card: The card is ejected after the transaction is complete.
8. End Session: The session ends, and the ATM goes back to an idle state.


Customer
ATM Machine
Bank
Insert ATM Card
Validate ATM Card
invalid
Eject Card
valid
Take Card
Enter PIN
Authorize PIN
valid pin
invalid pin
Enter Amount
Check Balance
Balance >= Amount
Take Money from Slot
Debit Account
Balance < Amount
Show Balance
Take Card
Eject Card

Component diagram

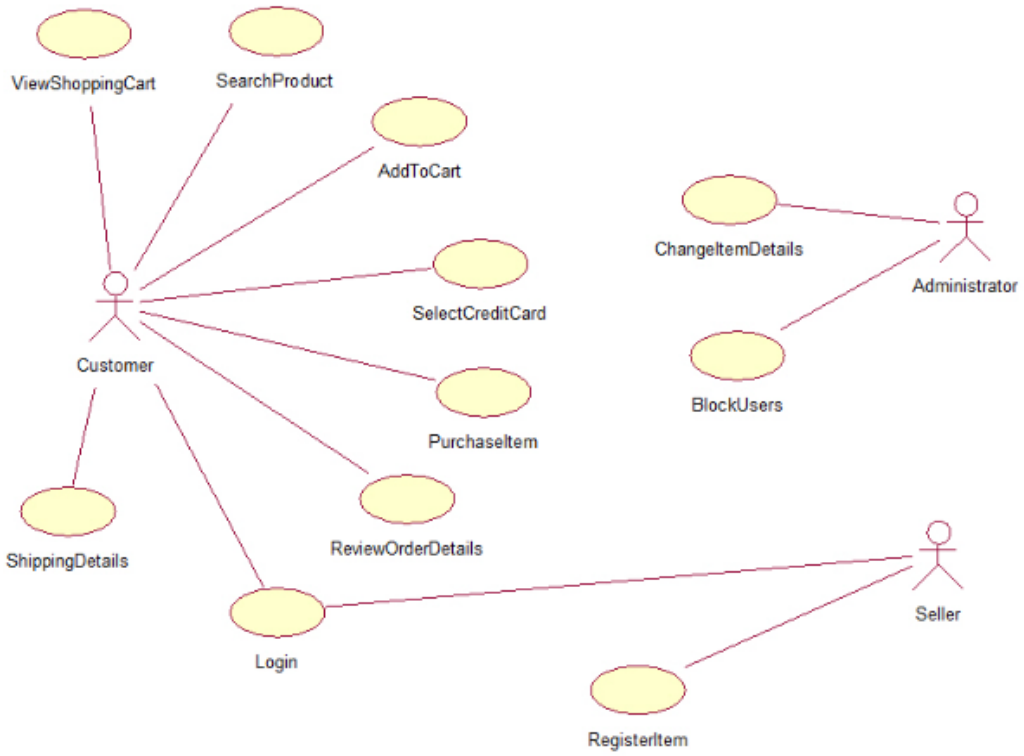


Online Shopping website

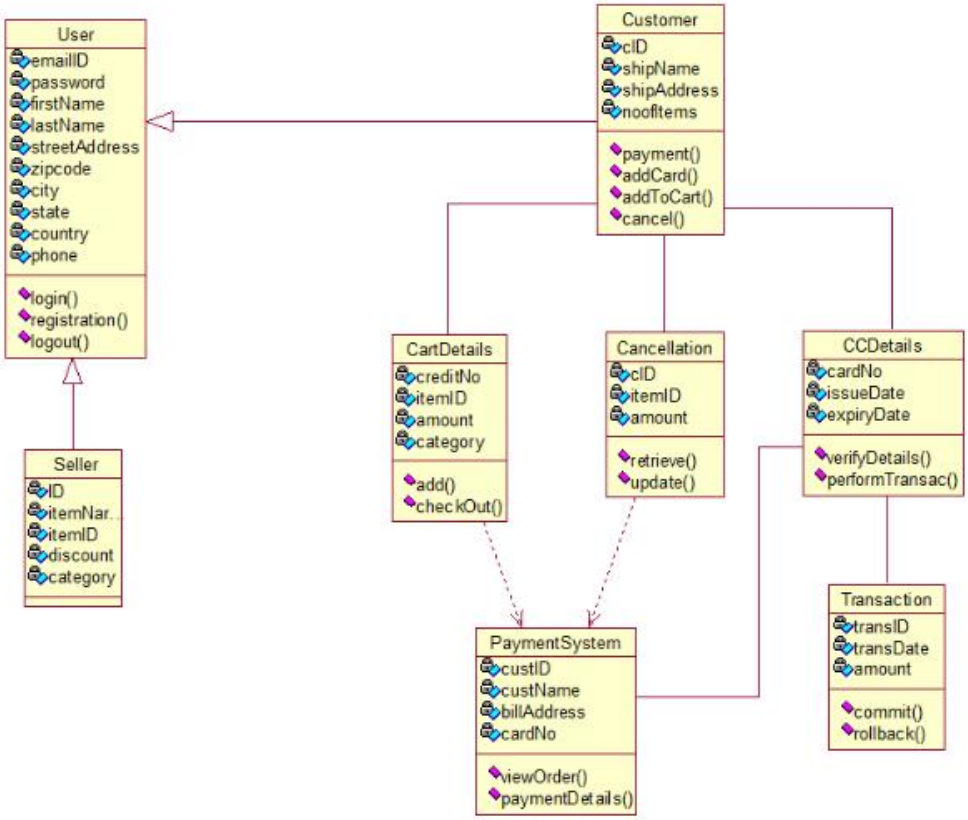
1. Use Case Diagram for Online Shopping Website
The **Use Case Diagram** shows the interactions between the **User** (or Customer) and the Online Shopping System.
Use Cases:

- Register/Login
- Browse Products
- View Product Details
- Add Product to Cart
- Remove Product from Cart
- Checkout
- Make Payment
- Track Order
- View Order History

Use Case Diagram



Class Diagram



2. Class Diagram for Online Shopping Website

The **Class Diagram** shows the structure of the system, with classes representing the main entities involved in the online shopping process.

Classes:

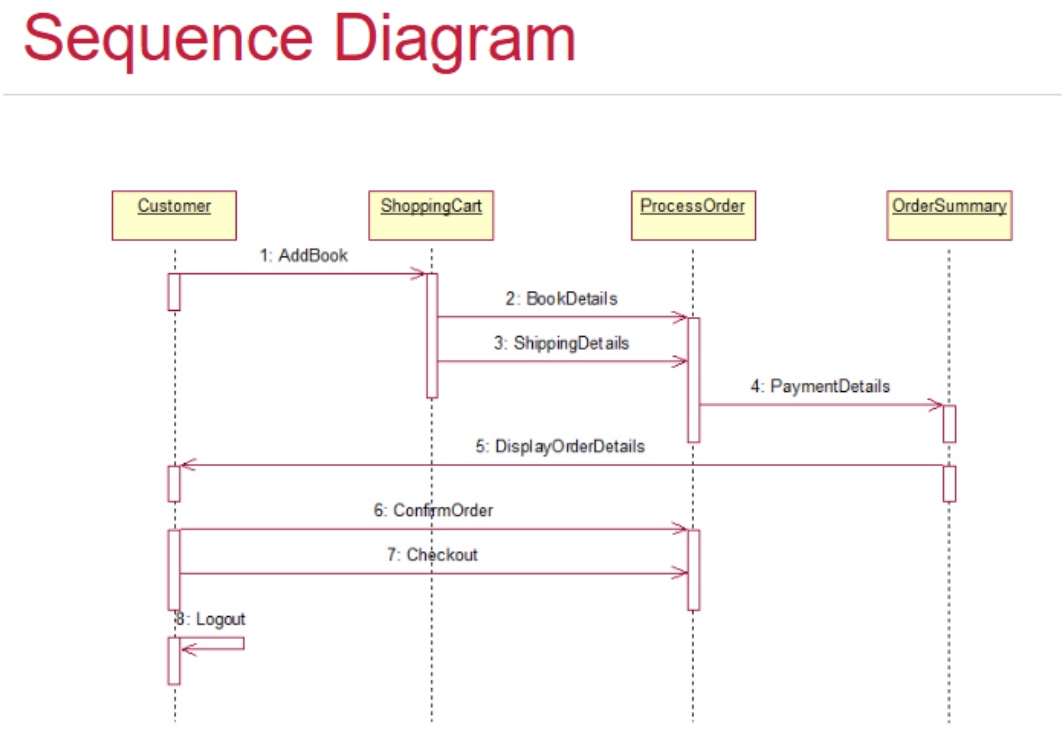
- **User**: Represents a registered user of the website.
- **Product**: Represents a product being sold on the site.
- **Cart**: Represents a shopping cart where products are added.
- **Order**: Represents an order made by the customer.
- **Payment**: Represents the payment process for an order.
- **Admin**: Represents the administrator who manages products and orders.

- **PaymentGateway**: Represents the third-party payment system integrated into the website.
- **Shipping**: Represents the shipping process for an order.

3. Sequence Diagram for Online Shopping Website
The **Sequence Diagram** models the interaction flow when a **customer buys a product**.
Objects:

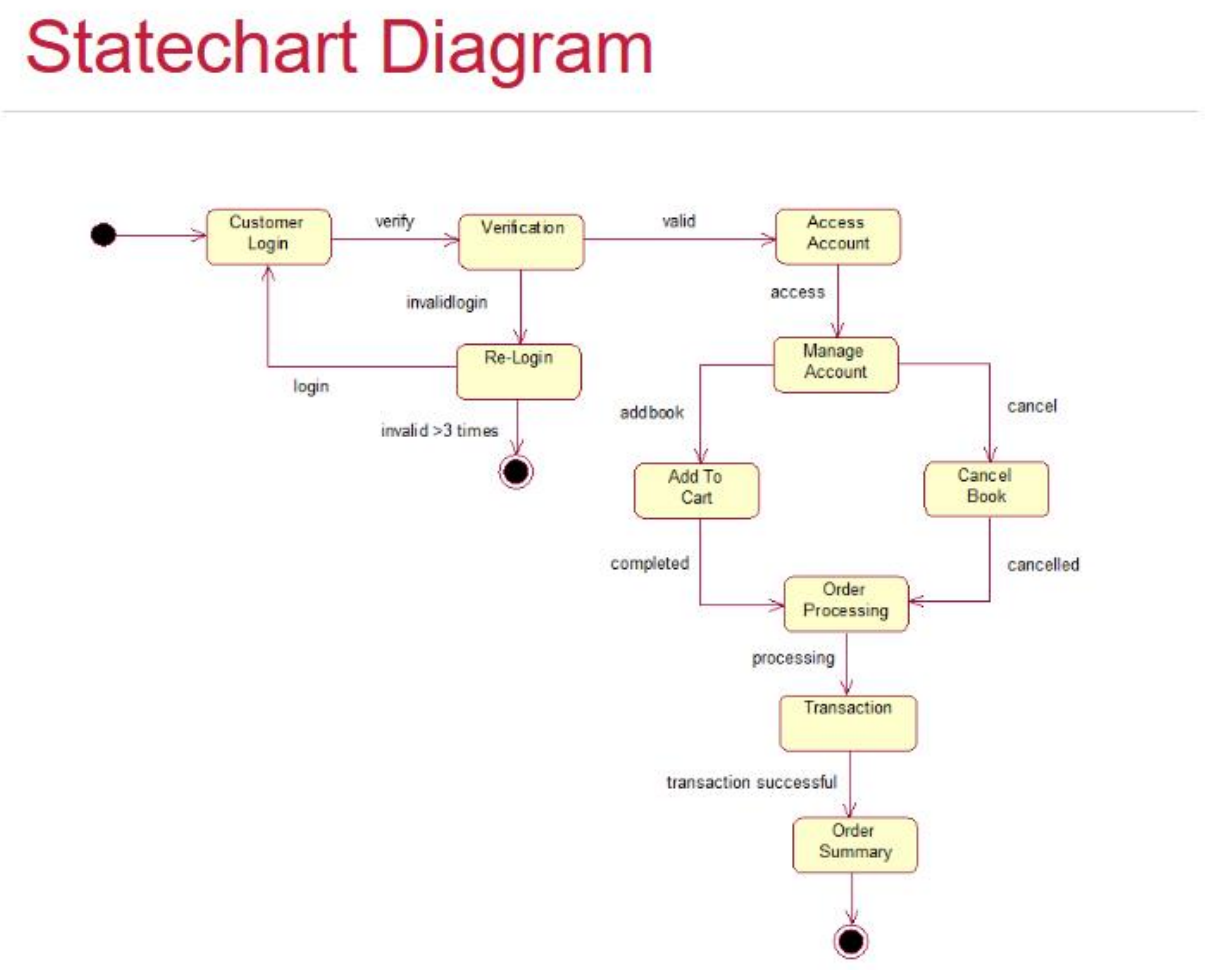
- **Customer**
- **Shopping Website**
- **Product**
- **Cart**
- **Order**
- **Payment Gateway**
- **Shipping**

Scenario: Customer purchases a product



4. State Machine Diagram for Online Shopping Website
The **State Machine Diagram** represents the possible states of an **Order** in the online shopping system.
States:

- **Browsing**: The user is browsing products.
- **Product Added to Cart**: The user adds a product to the cart.
- **Checkout**: The user proceeds to checkout.
- **Payment Pending**: The user is in the payment process.
- **Order Confirmed**: The payment is confirmed, and the order is processed.
- **Shipped**: The order is shipped to the customer.
- **Delivered**: The order has been delivered.



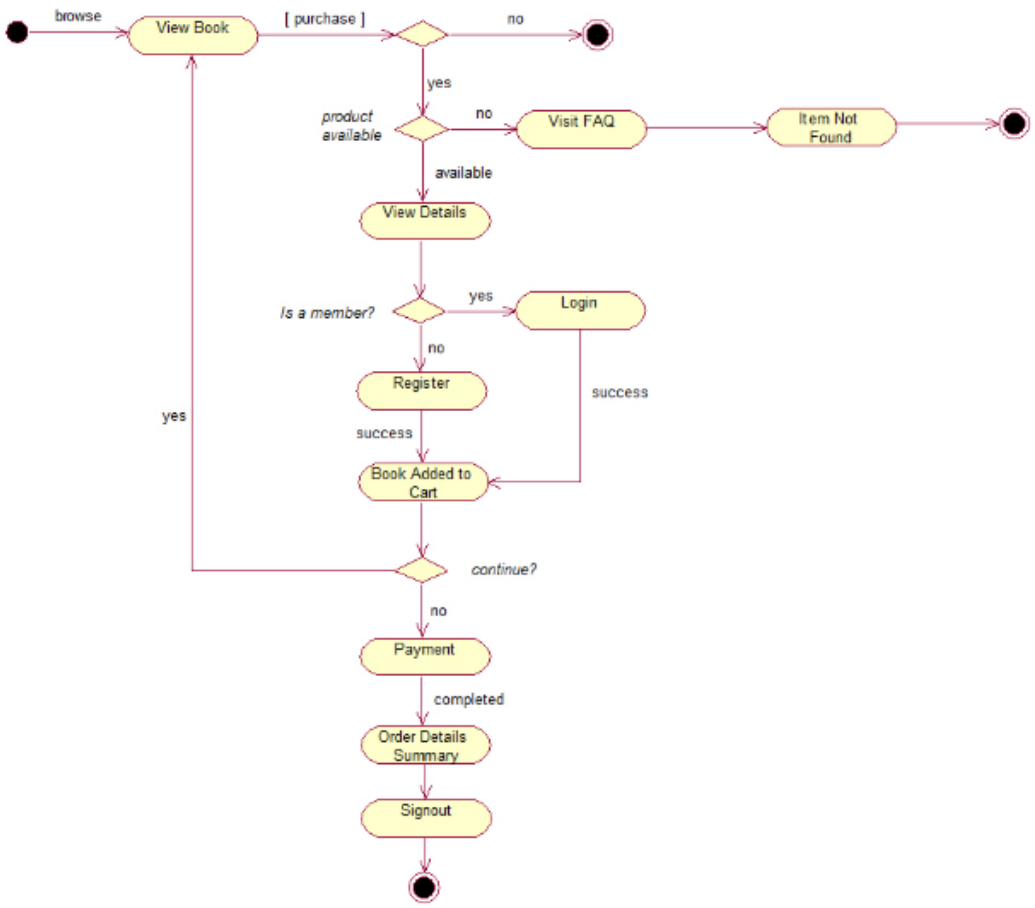
Activity Diagram for Online Shopping Website

The **Activity Diagram** represents the flow of actions the customer takes when shopping online. This captures the entire process, from logging in to placing an order and completing the payment.

Key Activities:

1. **Log in/Register**: The user logs into or registers on the website.
2. **Browse Products**: The user browses products in the catalog.
3. **View Product Details**: The user views more details about a product.
4. **Add Product to Cart**: The user adds selected items to the shopping cart.
5. **Proceed to Checkout**: The user proceeds to checkout.
6. **Enter Shipping Information**: The user provides shipping information.
7. **Make Payment**: The user enters payment details.
8. **Confirm Order**: The user confirms the order.
9. **Track Order**: The user tracks the status of their order.
10. **Logout**: The user logs out of the website.

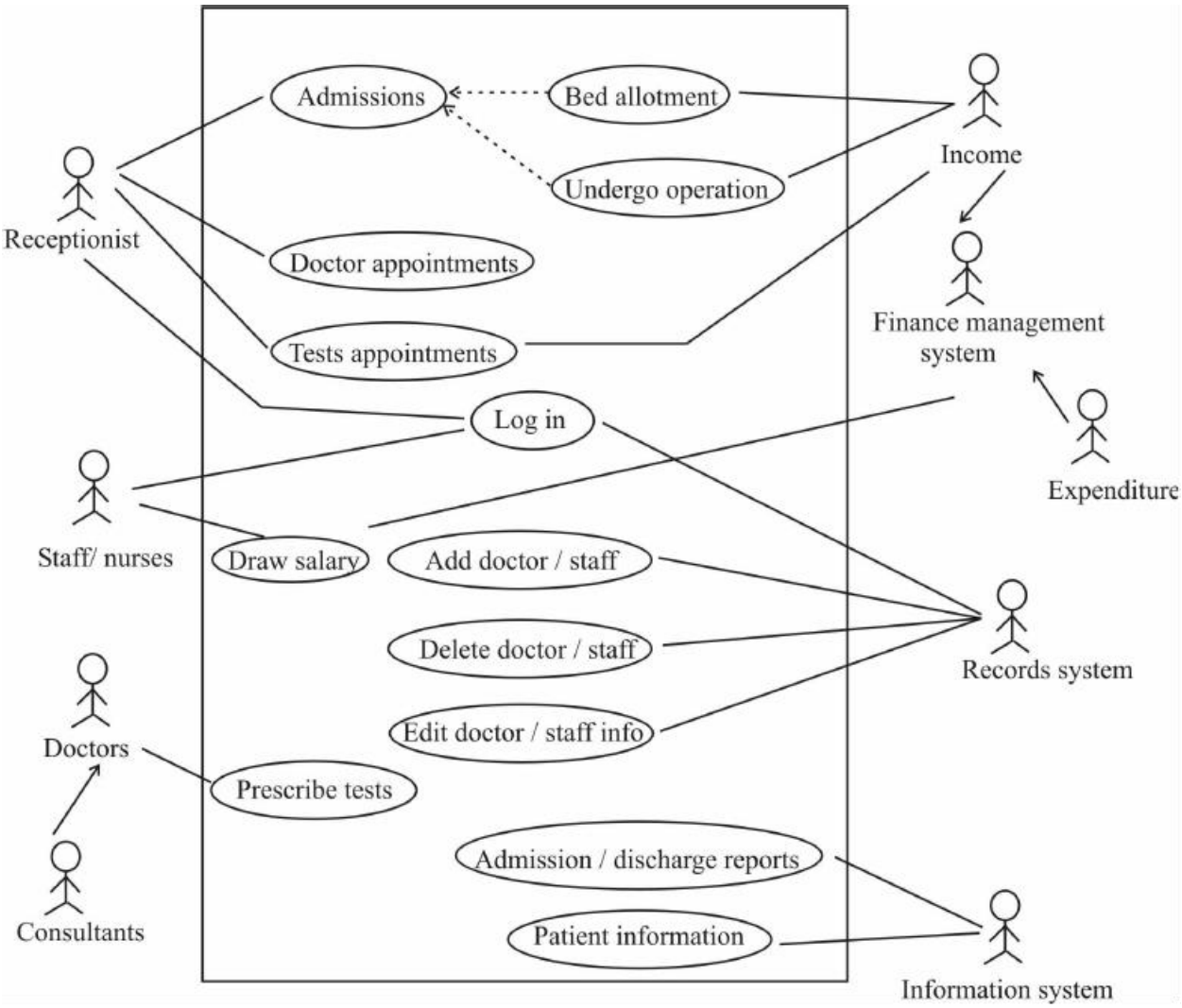
Activity Diagram



Hospital management system

1. Use Case Diagram for Hospital Management System
The **Use Case Diagram** shows how users interact with the hospital management system. The **actors** interact with the system through various use cases, such as patient registration, appointment scheduling, and generating bills.
Actors:

- **Patient**: The person seeking treatment.
- **Doctor**: The person providing medical care.
- **Nurse**: The healthcare worker assisting doctors.
- **Admin**: The hospital administrator who manages system operations.
- **Receptionist**: Handles appointments and patient check-in.
- **Payment Gateway**: Third-party system for payment processing.



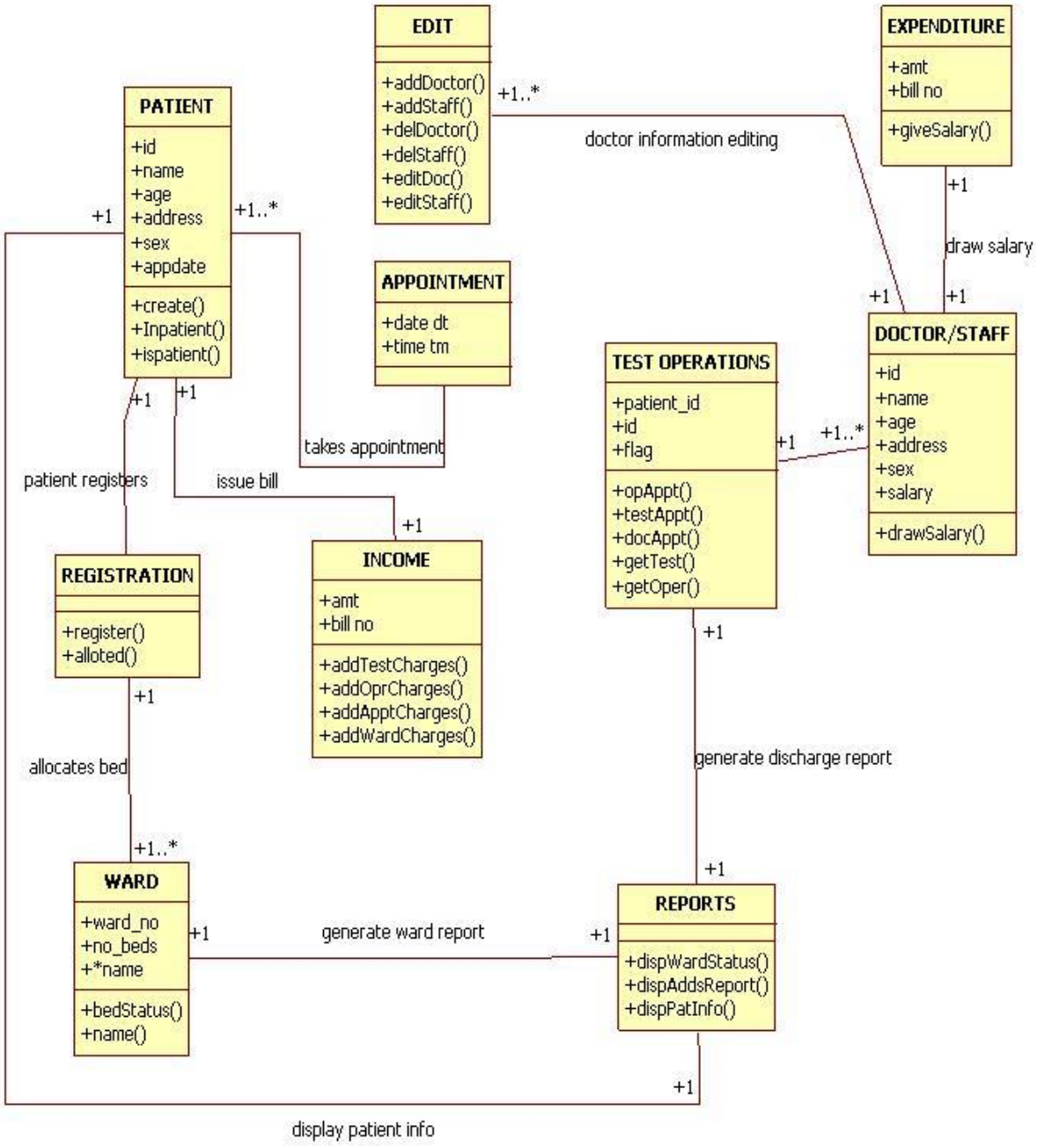
. Class Diagram for Hospital Management System

The **Class Diagram** represents the structure of the system, showing the relationships between key objects and classes like **Patient**, **Appointment**, **Doctor**, **Bill**, and **Prescription**.

Key Classes:

- **Patient**: Contains details about the patient like ID, name, medical history, etc.
- **Doctor**: Contains details about the doctor like specialization, schedule, etc.
- **Appointment**: Contains details about appointments including time, patient, doctor, and status.
- **Bill**: Contains details about charges, payments, and outstanding balances.
- **Prescription**: Contains details about prescribed medication for the patient.

- **Admin**: Manages hospital operations like patient records, doctor schedules, and bills.
- **PaymentGateway**: Handles payment processing.



3. Sequence Diagram for Hospital Management System

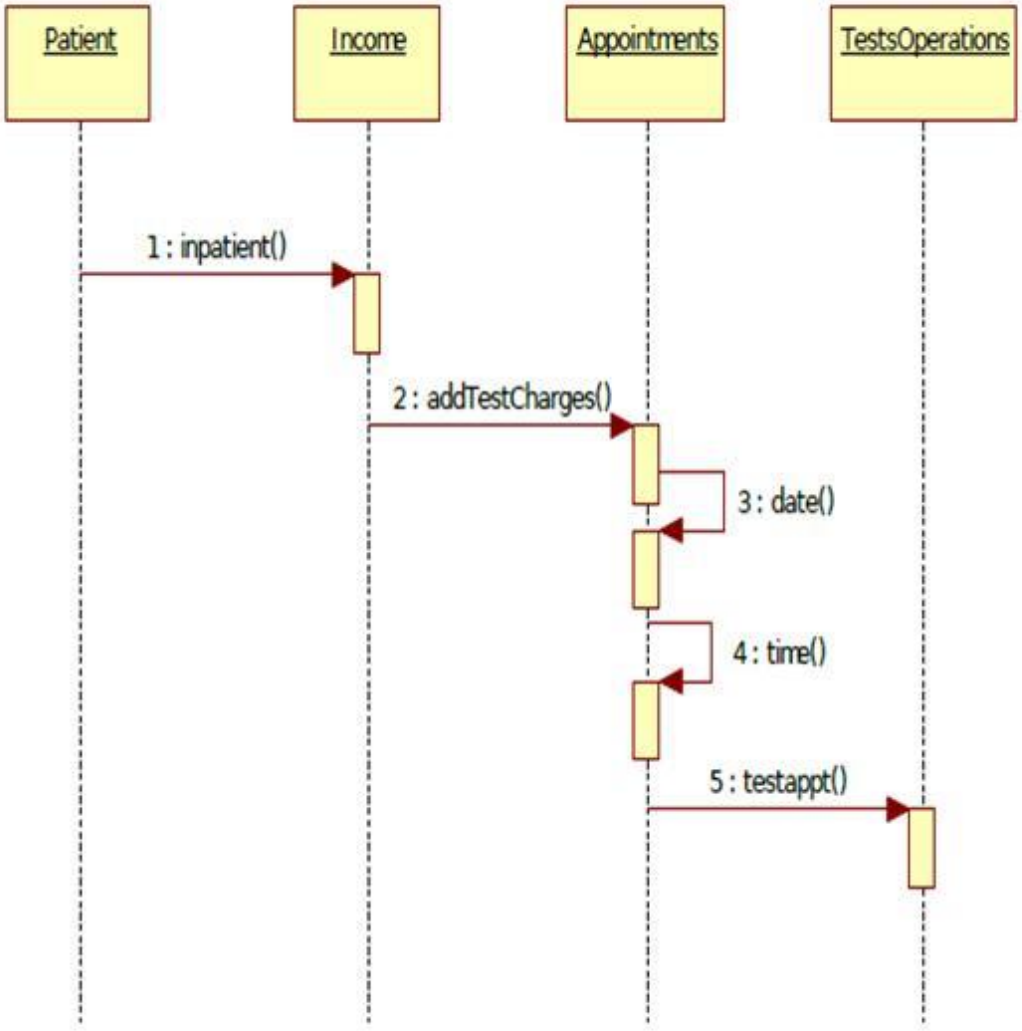
The **Sequence Diagram** models the interaction flow between objects during a **patient's appointment booking process**.

Objects:

- **Patient**
- **Hospital System**
- **Doctor**
- **Appointment**
- **Payment Gateway**

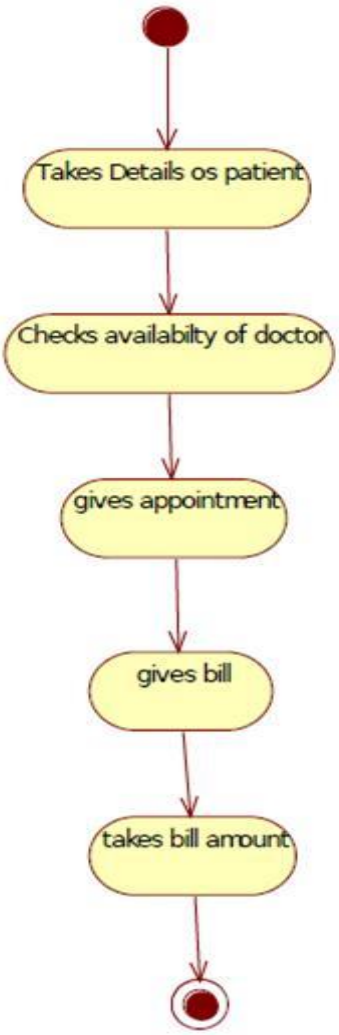
Scenario: A patient books an appointment and makes a payment

TESTS APPOINTMENTS:



. State Machine Diagram for Hospital Management System
The **State Machine Diagram** represents the states of an **Appointment** in the system.
States:

- **Scheduled**: The appointment is booked but not yet attended.
- **Attended**: The patient has seen the doctor.
- **Bill Generated**: The bill for the consultation and treatment has been generated.
- **Payment Pending**: The payment for the bill is pending.

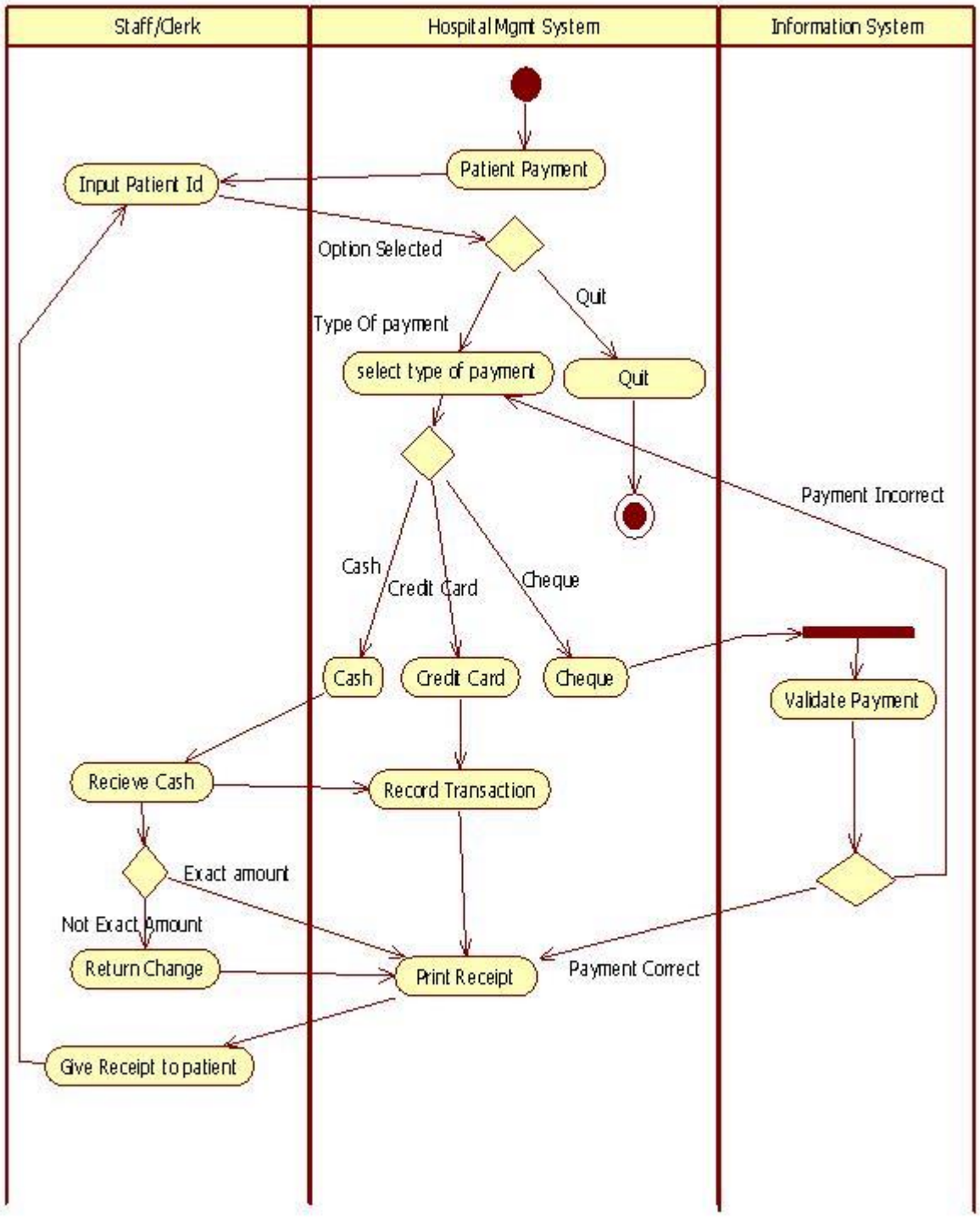


- **Paid**: The bill is paid.

5. Activity Diagram for Hospital Management System
The **Activity Diagram** represents the workflow of a patient interacting with the system, from registering to receiving treatment and making a payment.
Key Activities:

1. **Register/Log in**: The patient registers or logs into the system.
2. **Book an Appointment**: The patient selects a doctor and schedules an appointment.

3. **Check-in for Appointment**: The patient checks in on the appointment date.
4. **Consult the Doctor**: The doctor examines the patient.
5. **Prescribe Medication**: The doctor prescribes necessary treatment.
6. **Generate Bill**: The hospital generates the bill for consultation and treatment.
7. **Make Payment**: The patient pays the bill.
8. **Logout**: The patient logs out after the process is complete.


Staff/Clerk
Hospital Mgmt System
Information System
Patient Payment
Input Patient Id
Option Selected
Quit
Type Of payment
select type of payment
Quit
Payment Incorrect
Cash
Credit Card
Cheque
Cash
Credit Card
Cheque
Validate Payment
Recieve Cash
Record Transaction
Exact amount
Not Exact Amount
Return Change
Print Receipt
Payment Correct
Give Receipt to patient

Library Management System

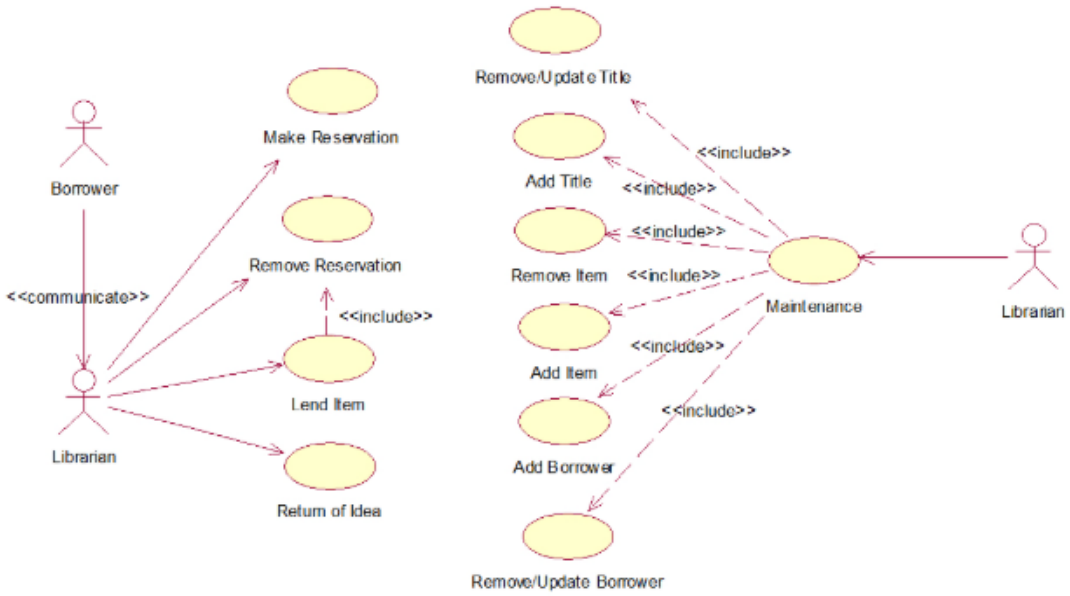
1. Use Case Diagram for Library Management System
The **Use Case Diagram** shows how different **actors** (users) interact with the **Library Management System**. The actors interact with the system for activities like registering, borrowing books, returning them, and checking their fines.
Actors:

- **Library User (Member)**: The person who borrows books.
- **Librarian**: The person who manages books, users, and fines.
- **System**: The automated system that manages the library.
- **Admin**: The system administrator who manages library system settings.

Use Cases:

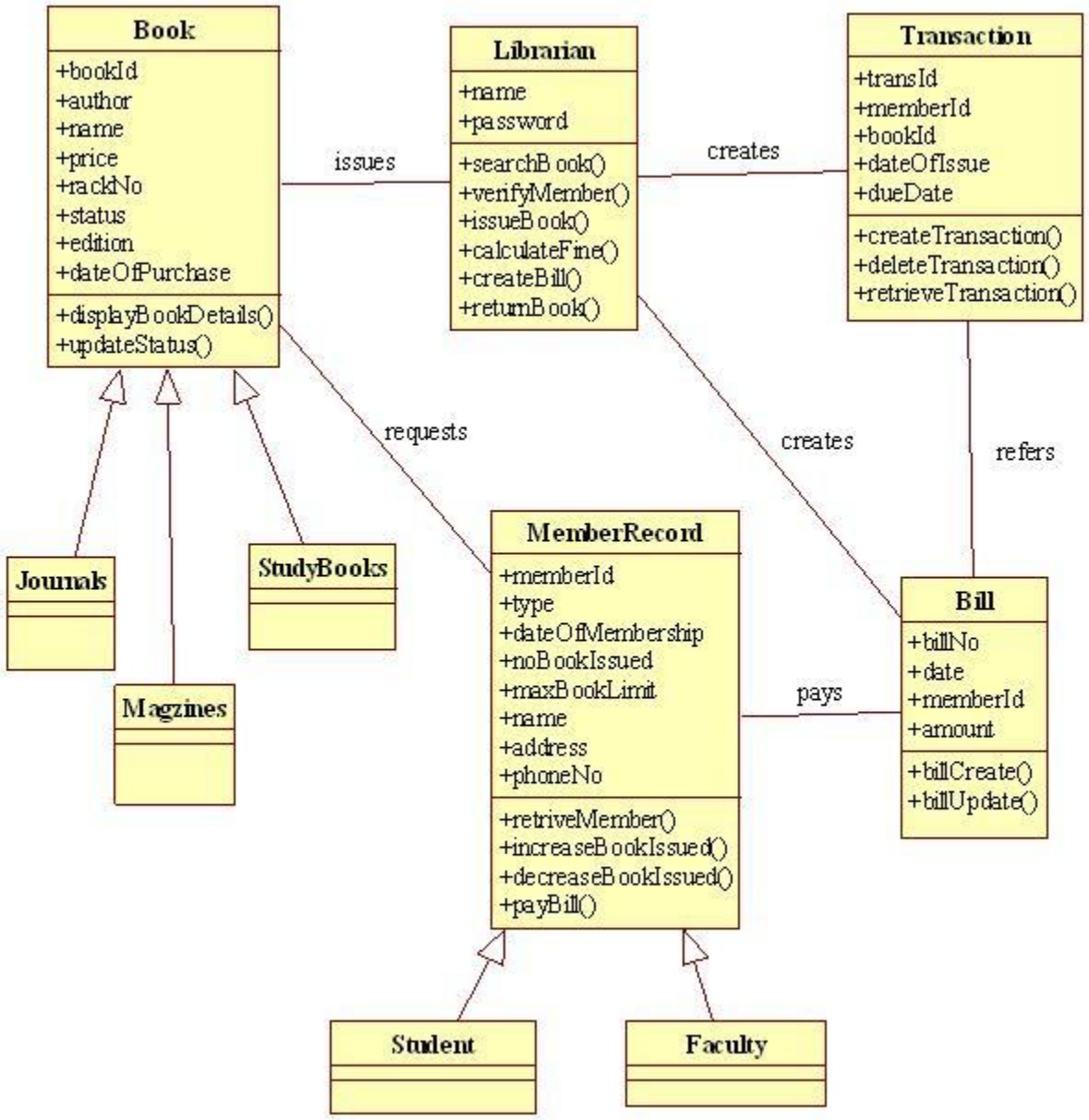
- Register as a new member
- Login to the system
- Search for books
- Borrow books
- Return books
- View borrowed books
- Pay fines
- Add/Remove books (for Librarian)
- View member records (for Librarian)

Use case diagram



2. Class Diagram for Library Management System
The **Class Diagram** represents the structure of the system, showing key classes like **User**, **Book**, **Loan**, **Fine**, **Librarian**, and **Admin**.
Key Classes:

- **User**: Represents a library member with attributes like name, ID, and borrowed books.
- **Book**: Represents a book with details such as title, author, and ISBN.
- **Loan**: Represents the loan details (book borrow records) of a user.
- **Fine**: Represents any overdue fines a user might incur.
- **Librarian**: The librarian who manages book inventory, user records, and fines.
- **Admin**: The administrator responsible for managing the system configuration.



3. Sequence Diagram for Library Management System

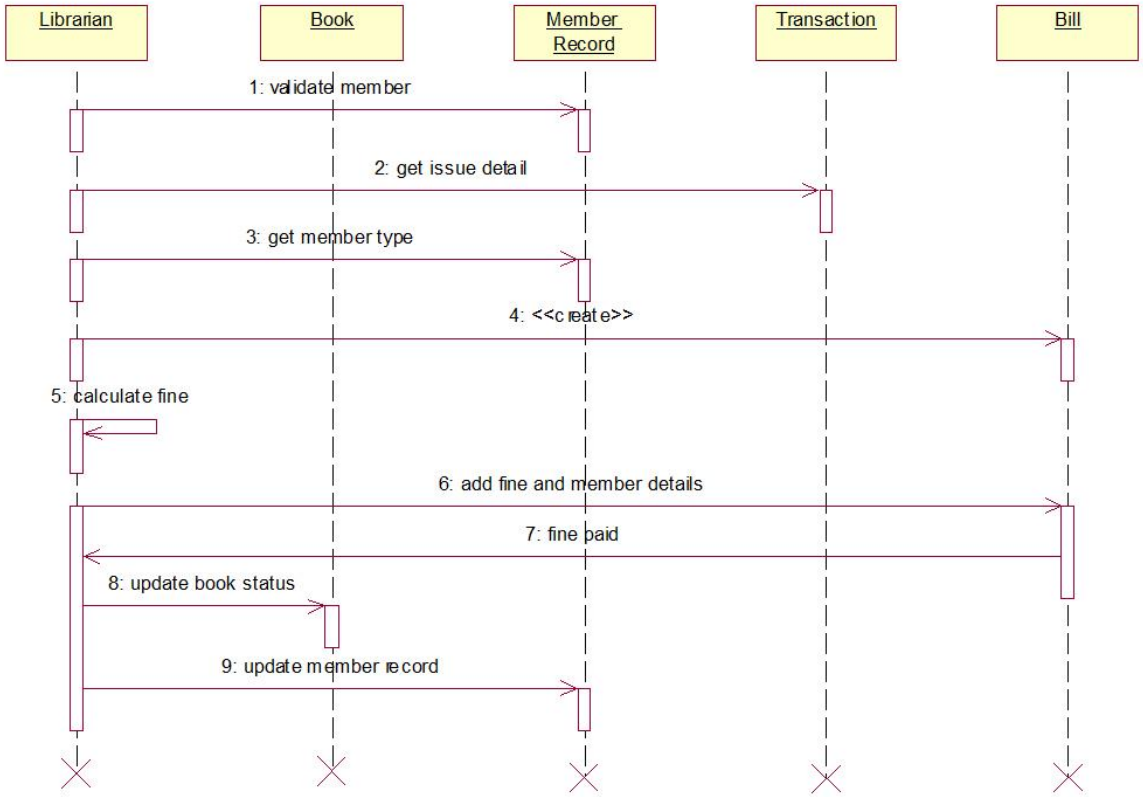
The **Sequence Diagram** models the interaction between a **user** and the system during the process of borrowing a book.

Objects:

- **User**
- **Library System**
- **Book**

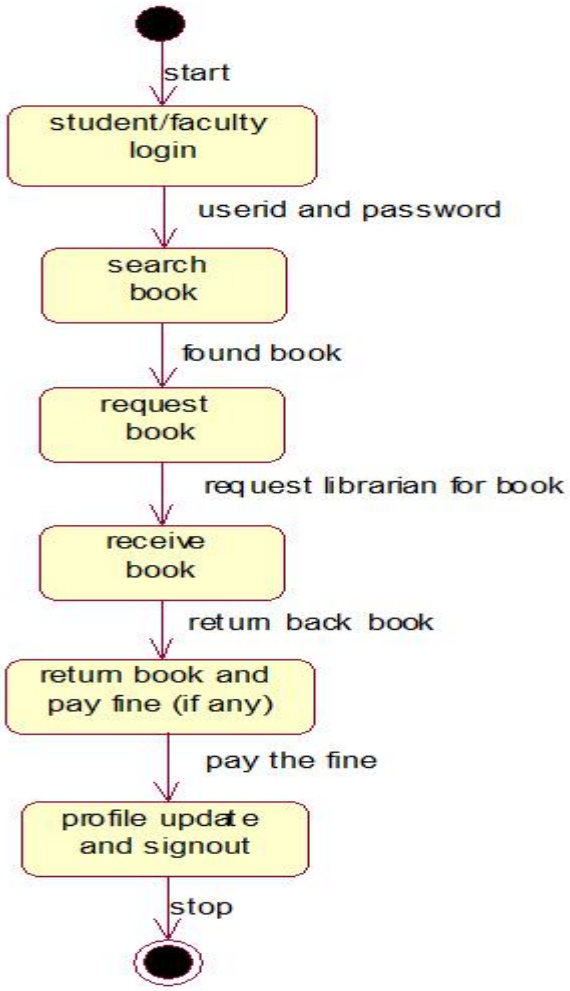
- **Loan**

Scenario: A user borrows a book.



4. State Machine Diagram for Library Management System
The **State Machine Diagram** shows the different states of a **Book** in the system. A book can be in multiple states, such as **Available**, **Borrowed**, and **Overdue**.
States:

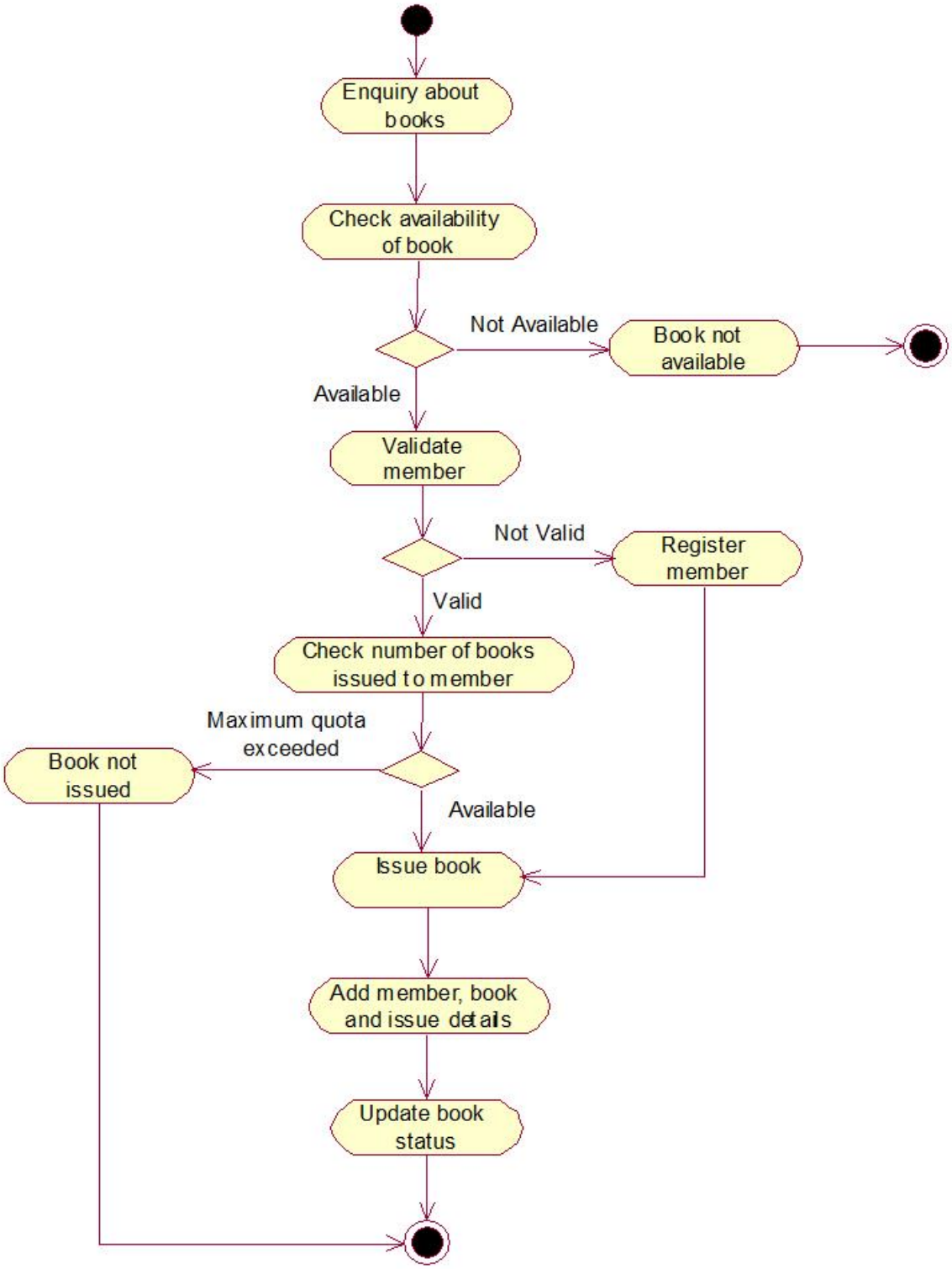
- **Available**: The book is available for borrowing.
- **Borrowed**: The book has been borrowed by a user.
- **Returned**: The book has been returned.
- **Overdue**: The book is past the return date and fines may be incurred.



Activity Diagram for Library Management System
The **Activity Diagram** models the flow of activities when a user borrows a book from the library.
Key Activities:

1. **Login/Register**: The user logs into or registers with the system.
2. **Search for Books**: The user searches for a book.
3. **View Book Details**: The user checks the details of the book (availability, author, etc.).
4. **Borrow Book**: The user borrows the book if it's available.
5. **Loan Book**: The system updates the loan record.
6. **Confirm Borrowing**: The user confirms the borrowing of the book.

7. **Return Book**: The user returns the book after the due date.
8. **Pay Fine**: If the book is returned late, the user pays the fine.


Enquiry about books
Check availability of book
Not Available
Book not available
Available
Validate member
Not Valid
Register member
Valid
Check number of books issued to member
Maximum quota exceeded
Book not issued
Available
Issue book
Add member, book and issue details
Update book status