

B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



Lab Record

Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SHREYANSH SETHIYA

1BM22CS269

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Mar-June 2024

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING



CERTIFICATE

This is to certify that the Object-Oriented Modelling (23CS5PCOOM) laboratory has been carried out by **Shreyansh Sethiya**(1BM22CS269) during the 5th Semester Oct24-Jan2025.

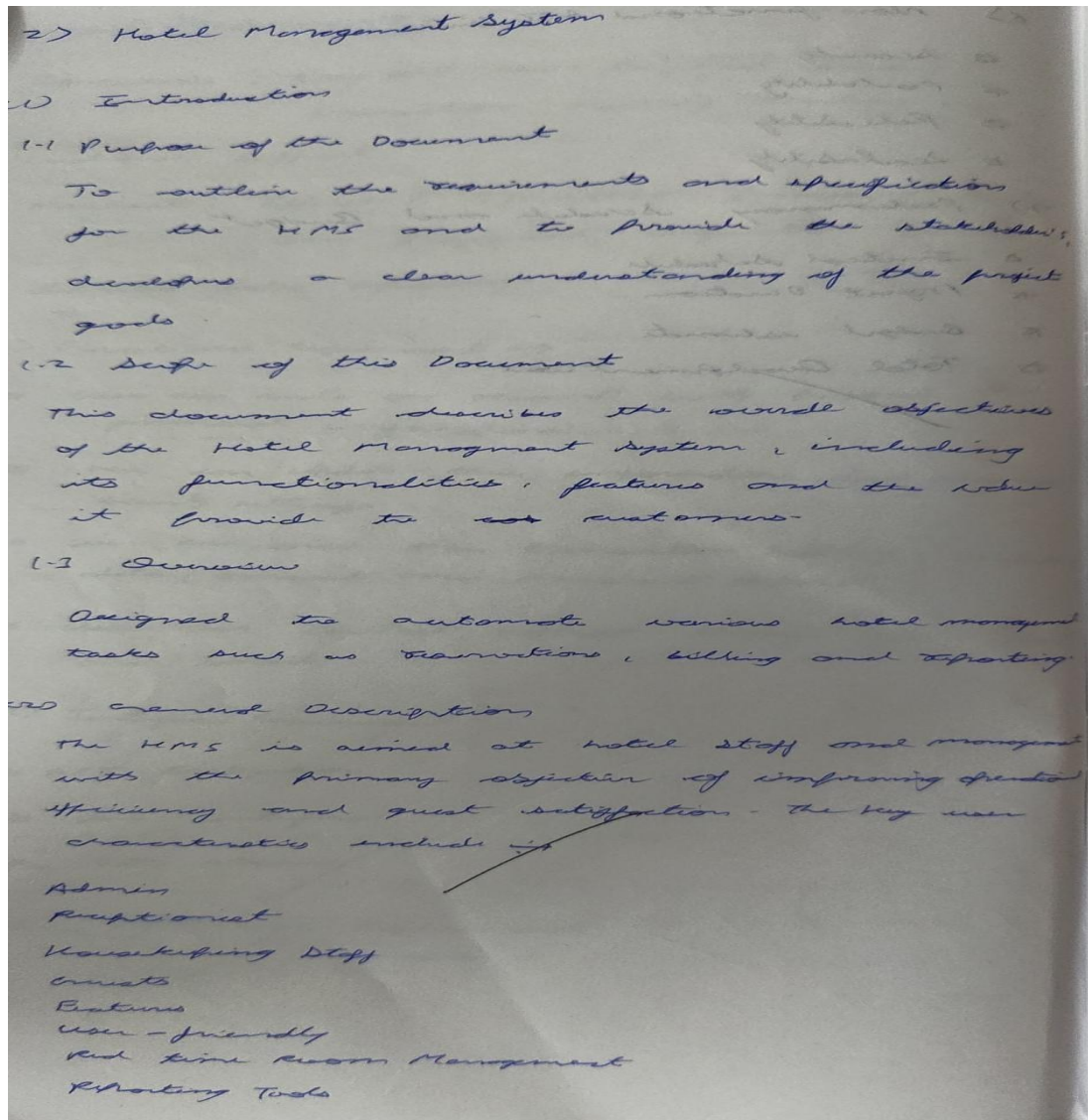
Signature of the Faculty Incharge:

Radhika A D
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

1. Hotel Management System
2. Credit Card Processing
3. Library Management System
4. Stock Maintenance System
5. Passport Automation System

1. Hotel Management System



Software Requirement Specification

(3) Functional Requirement

User Registration and Authentication

Admin can add, update and delete room information

Guests can search and book rooms based on availability

Reservations can be modified or cancelled.

Confirmation emails are sent upon booking.

Receptionists can check guests in and out

Invoices are generated upon check-out

System should support multiple payment methods and generate invoices.

Staff can update room cleanliness status and report maintain issues.

Admin can generate reports on occupancy rates and revenue.

(4) Interface Requirement

User Interface

Database interface

Email Interface

(5) Performance Requirements

Response time

Concurrency

Data Processing.

(6) Design Constraints

Technology Stack: The system must be built using specific technologies.

Database: use of a relational Database

(7) Non-functional Attributes

Security

Reliability

Accessibility

Portability

(8) Preliminary Schedule and Budget

Estimated to be six months

Requirement Gathering : 4 weeks

Design : 4 weeks

Development : 10 weeks

Deployment : 2 weeks

Estimated Budget

Budget for the development of HRMS is approx
\$100,000 covering personnel costs, software
licenses and infrastructure.

Class Diagram

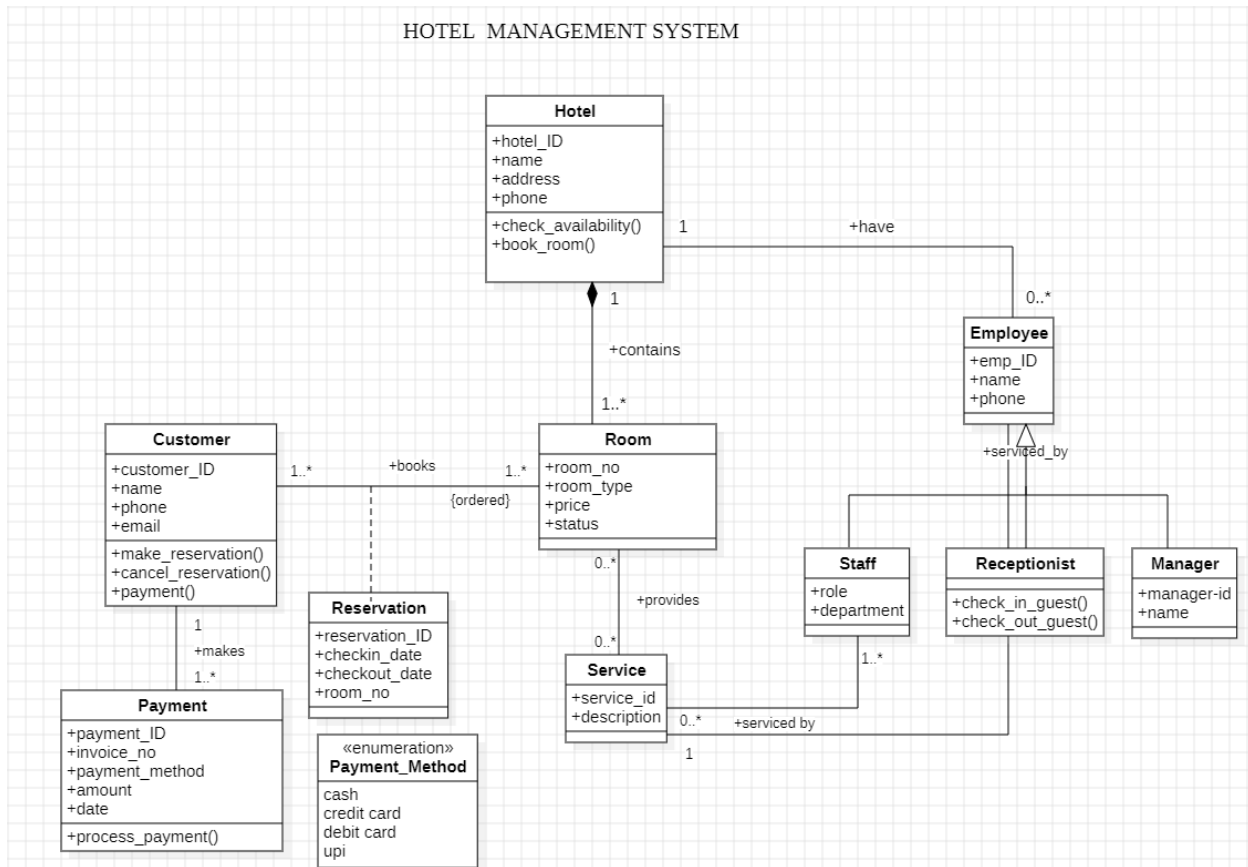


Fig1.1 Hotel Management System - Class Diagram

The diagram represents a hotel management system. It showcases the relationships between various entities such as Customer, Hotel, Room, Reservation, Payment, Service, and Staff. The diagram defines the attributes and operations associated with each entity, such as making a reservation, checking in/out guests, processing payments, etc. It also depicts the relationships between these entities, including one-to-one, one-to-many, and many-to-many relationships. For example, a customer can make multiple reservations, each reservation is associated with a specific room, and different types of staff members can be involved in various services. The diagram provides a comprehensive overview of the system's structure and interactions.

State Diagram

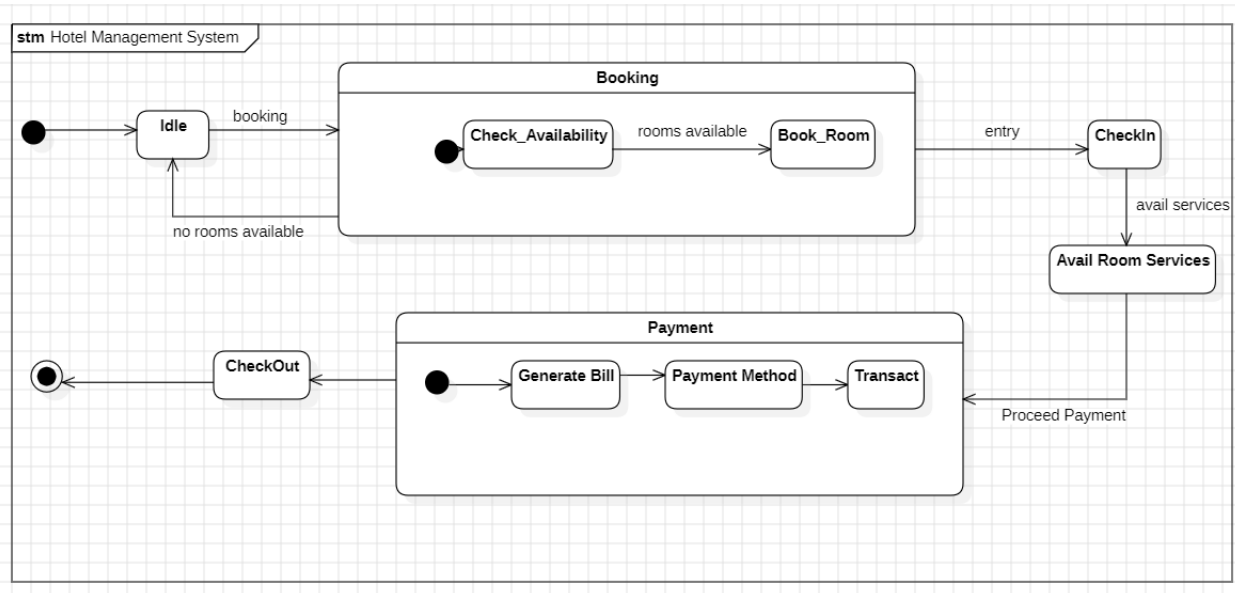


Fig1.2 Hotel Management System - State Diagram

The hotel management system state machine models the operational workflow of a hotel, transitioning through distinct states triggered by specific events. It begins in an Idle state, waiting for booking requests. Upon receiving a request, it transitions to Check_Availability to verify room availability. If rooms are available, it moves to Book_Room, confirming the booking; otherwise, it returns to Idle. Once booked, the customer proceeds to Checkin, after which they can avail services in Avail Room Services. The CheckOut state initiates upon the customer's departure, followed by Generate Bill to prepare their bill. In Payment Method, the customer selects how to pay, leading to Transact, where the payment is processed. Each state and transition ensures smooth and sequential operation of the system, ensuring efficiency and clarity in hotel management.

Use Case Diagram

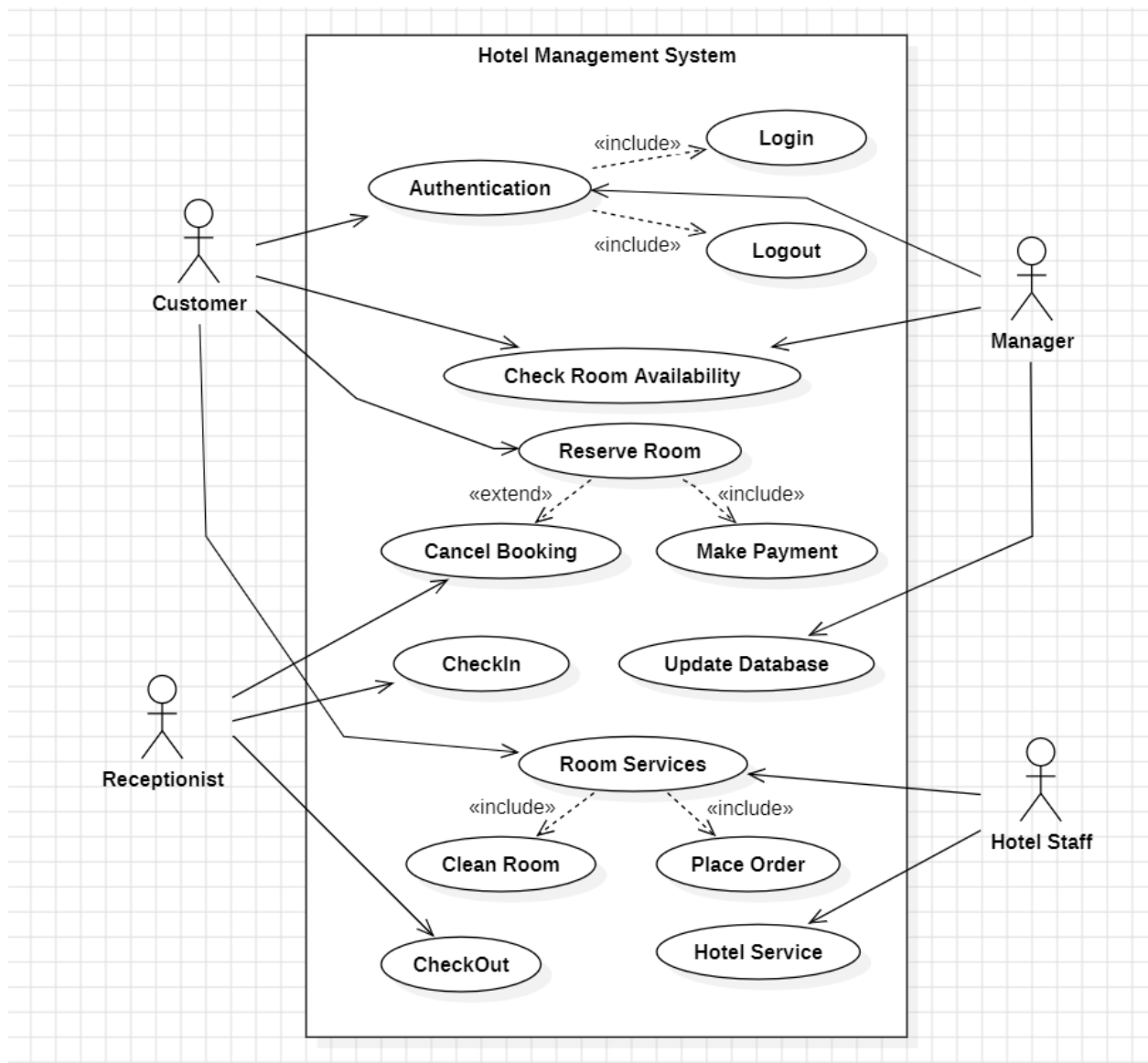


Fig1.3 Hotel Management System - Use Case Diagram

The diagram represents a Use Case Diagram for a Hotel Management System, showcasing various interactions between users (actors) and system functionalities. The primary actors include Customer, Manager, Receptionist, and Hotel Staff. Key use cases are grouped under the system, such as Authentication (which includes login and logout), Check Room Availability, Reserve Room (extended by Cancel Booking and including Make Payment), Check-In, and Room Services (further including cleaning, placing orders, and other hotel services). The diagram emphasizes the relationships and interactions among actors and system processes,

demonstrating how each user contributes to the system's operations. For example, the Manager and Receptionist oversee updates and reservations, while Hotel Staff handle room services.

Sequence Diagram

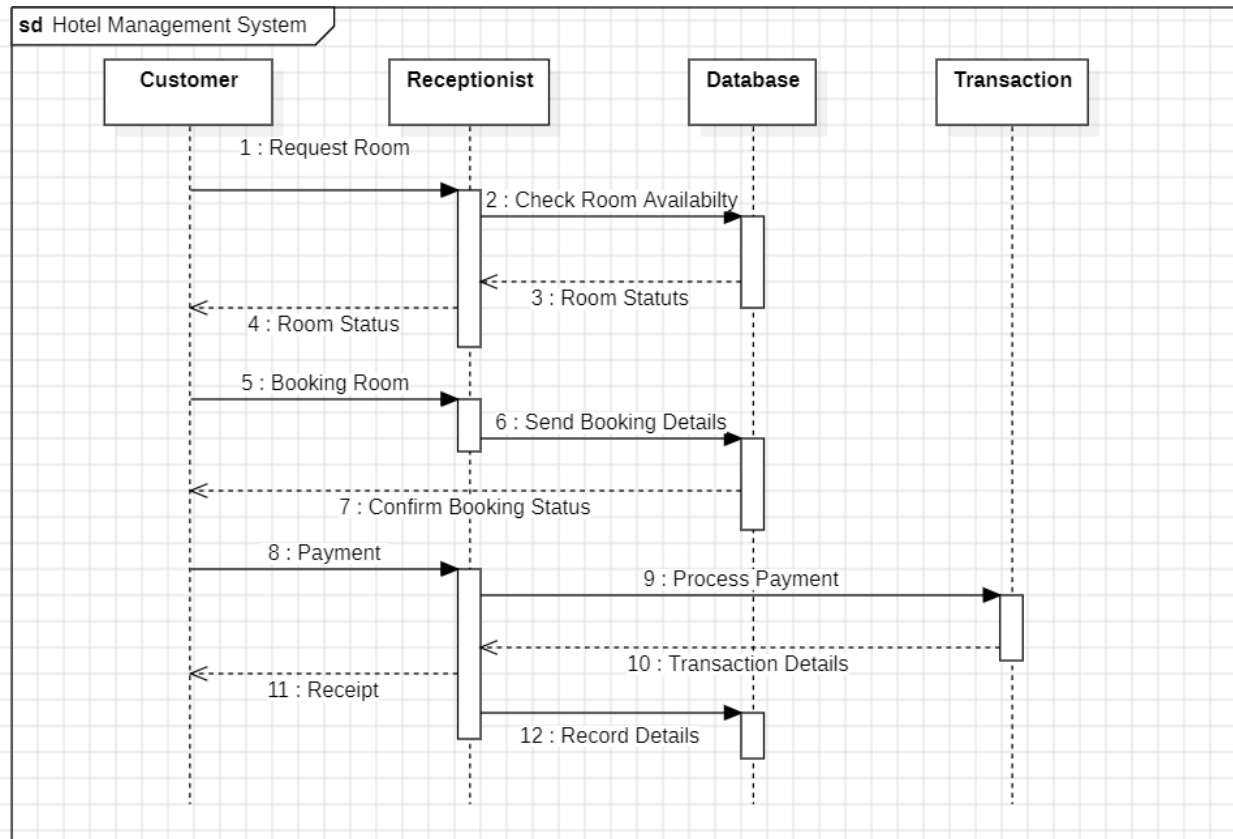


Fig1.4 Hotel Management System - Sequence Diagram

The sequence diagram illustrates the process of booking a room at a hotel. The customer initiates the process by requesting a room. The receptionist then checks the availability of the room in the database and returns the status to the customer. If the room is available, the customer can book the room. The receptionist sends the booking details to the database and confirms the booking status to the customer. The customer then makes the payment, and the transaction is processed by the database. Finally, the customer receives a receipt, and the database records the transaction details.

Activity Diagram

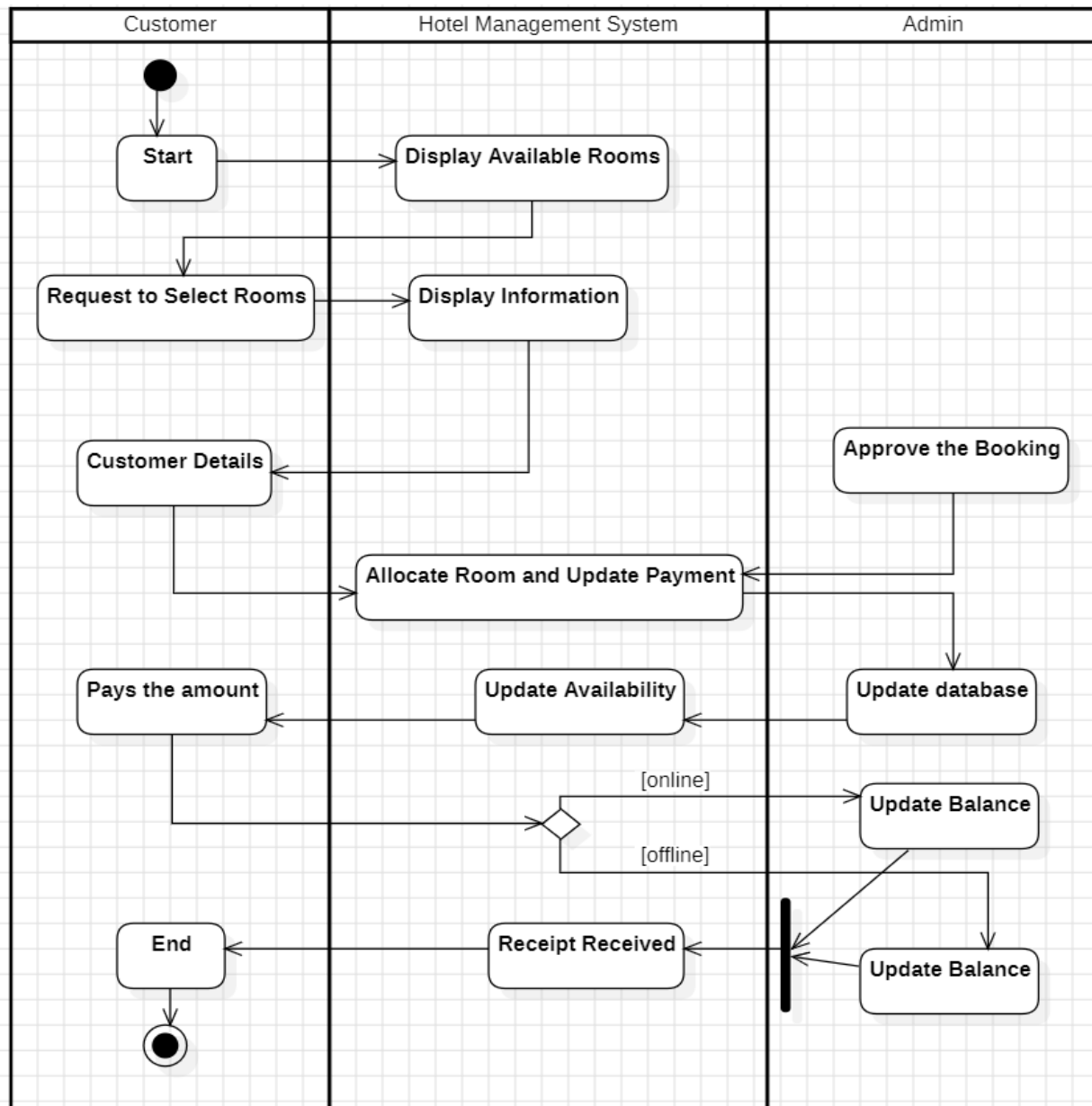
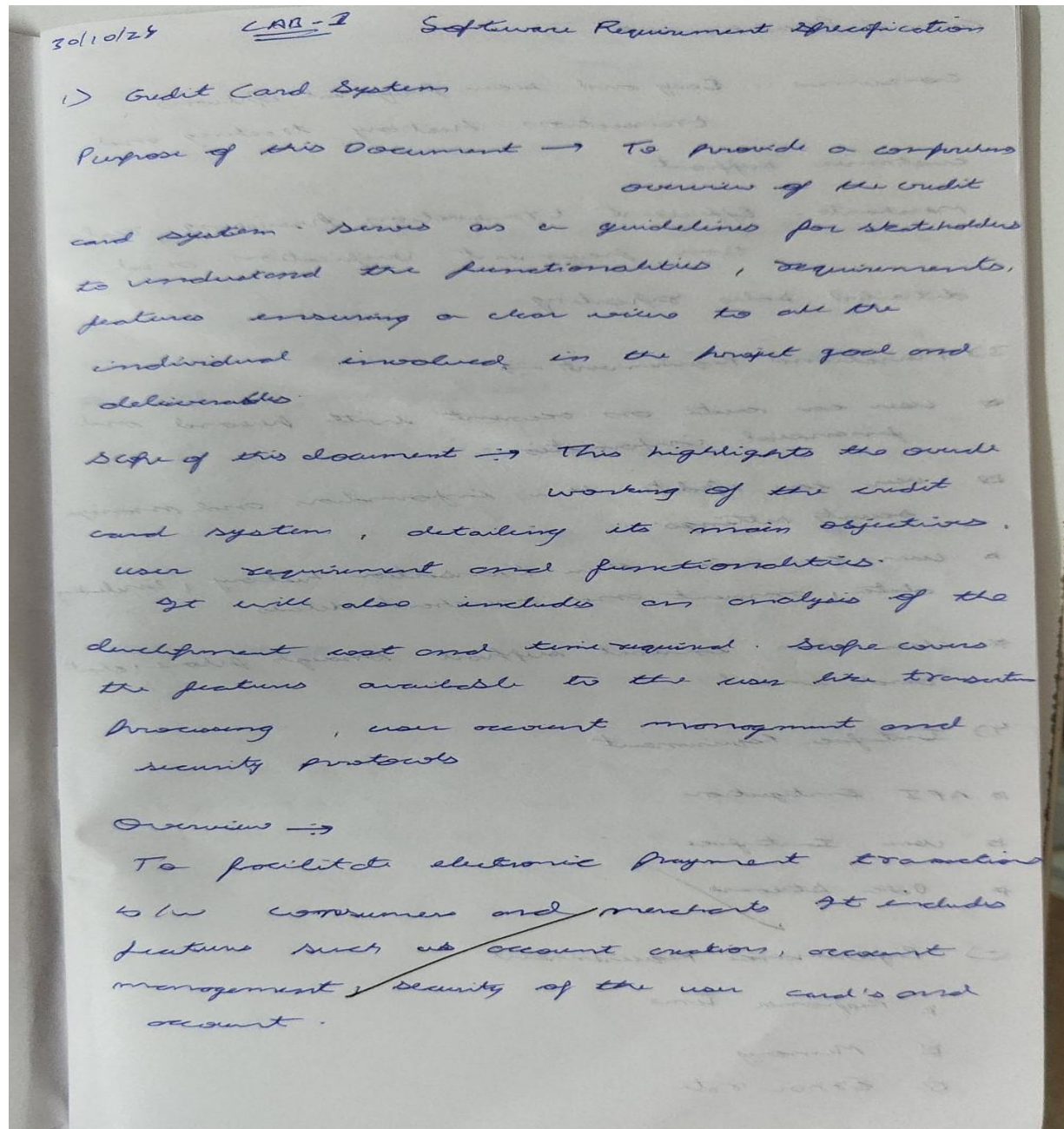


Fig1.5 Hotel Management System - Activity Diagram

The activity diagram illustrates the process of booking a room at a hotel. The customer starts by requesting to select rooms, and the system displays available rooms and their information. The customer then provides their details and selects a room. The admin approves the booking, and the system allocates the room and updates the payment. The customer pays the amount, and the

system updates the room availability and balance. Finally, the customer receives a receipt, and the system updates the balance.

2. Credit Card Processing System



Software Requirement Specification

General Descriptions →

Consumer: Easy and secure payment options,
transaction history tracking and
customer support

Merchant: Efficient transaction processing, real-
time payment verification and
detailed sales reporting.

3) Functional Requirement →

- * User can create an account with personal and financial information
- * User can update their information and manage security settings
- * User can view their transaction history, including date, amount and merchant details
- * User can contact support through phone, chat and email

4) Interface Requirement

- * API Integration
- * User Interface
- * Data Storage

5) Performance Requirement

- * Response Time
- * Memory
- * Error Rate

6) Non-functional Attributes

- * Security
- * Portability
- * Reliability
- * Scalability

7) Preliminary Schedule and Budget

- * Initial Schedule
- * Project Duration
- * Budget estimate
- * Total Development Cost

Class Diagram

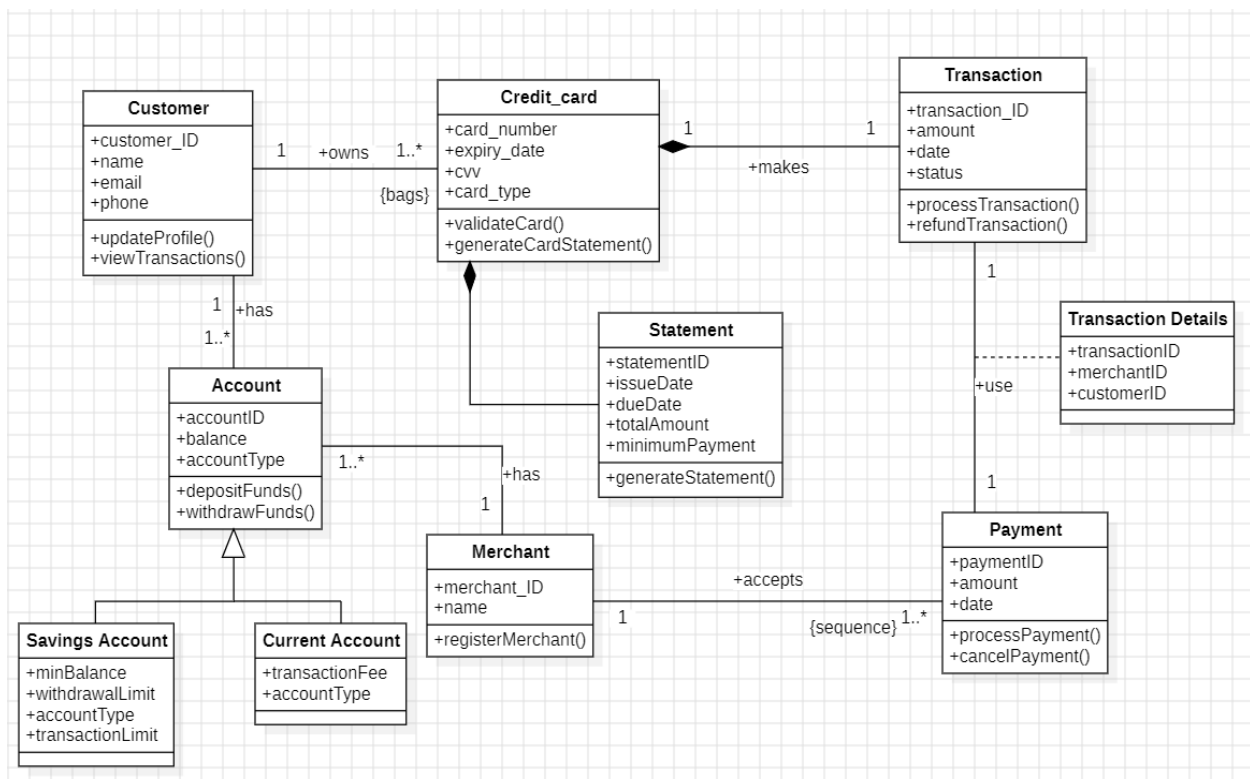


Fig 2.1 Credit Card Processing System - Class Diagram

The class diagram represents a credit card processing system. Customers own one or more Credit Cards, which are used to perform Transactions. Each credit card is validated and associated with a Statement that includes payment details like total amount and due date. Accounts (Savings or Current) store the customer's funds and enable deposits and withdrawals. Merchants register to accept payments, and payments are linked to Transaction Details, specifying the customer and merchant involved. Key functionalities include processing and refunding transactions, validating credit cards, generating statements, and updating customer profiles.

State Diagram

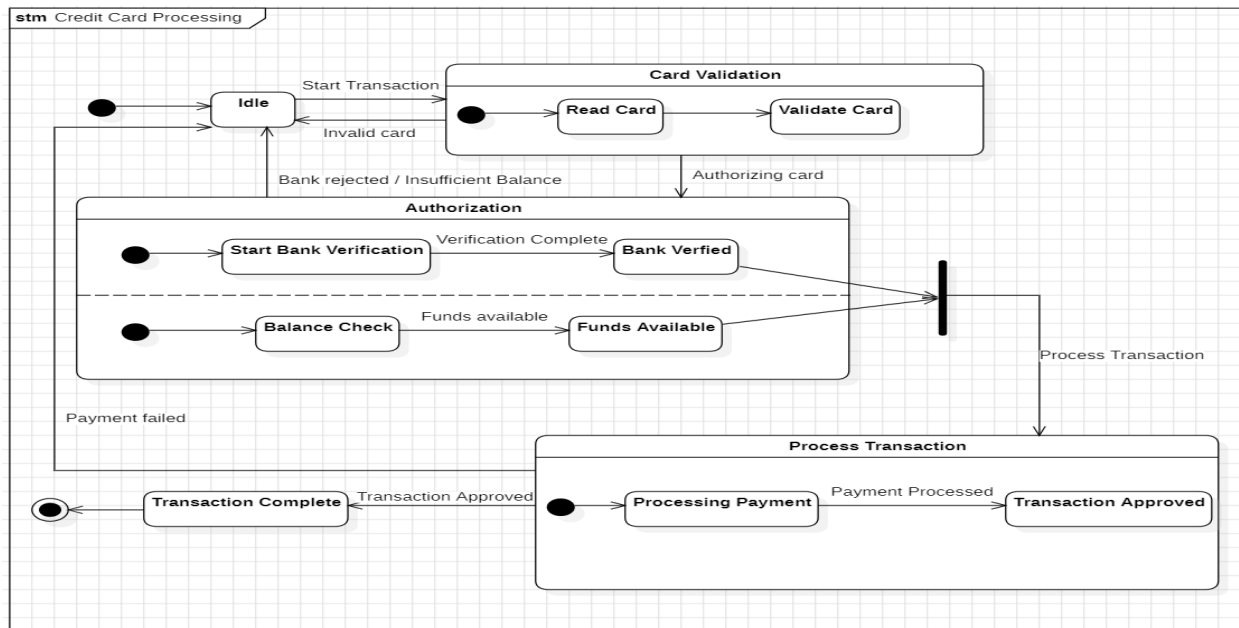


Fig 2.1 Credit Card Processing System - State Diagram

The state diagram illustrates the process of a credit card transaction. The system starts in an idle state and transitions to the "Read Card" state when a transaction is initiated. The card is then validated, and if it is invalid, the transaction is rejected. If the card is valid, the system moves to the "Authorization" state and verifies the card with the bank. If the card is verified and the funds are available, the system proceeds to the "Process Transaction" state and completes the transaction. If the card is not verified or there are insufficient funds, the transaction fails.

Use Case Diagram

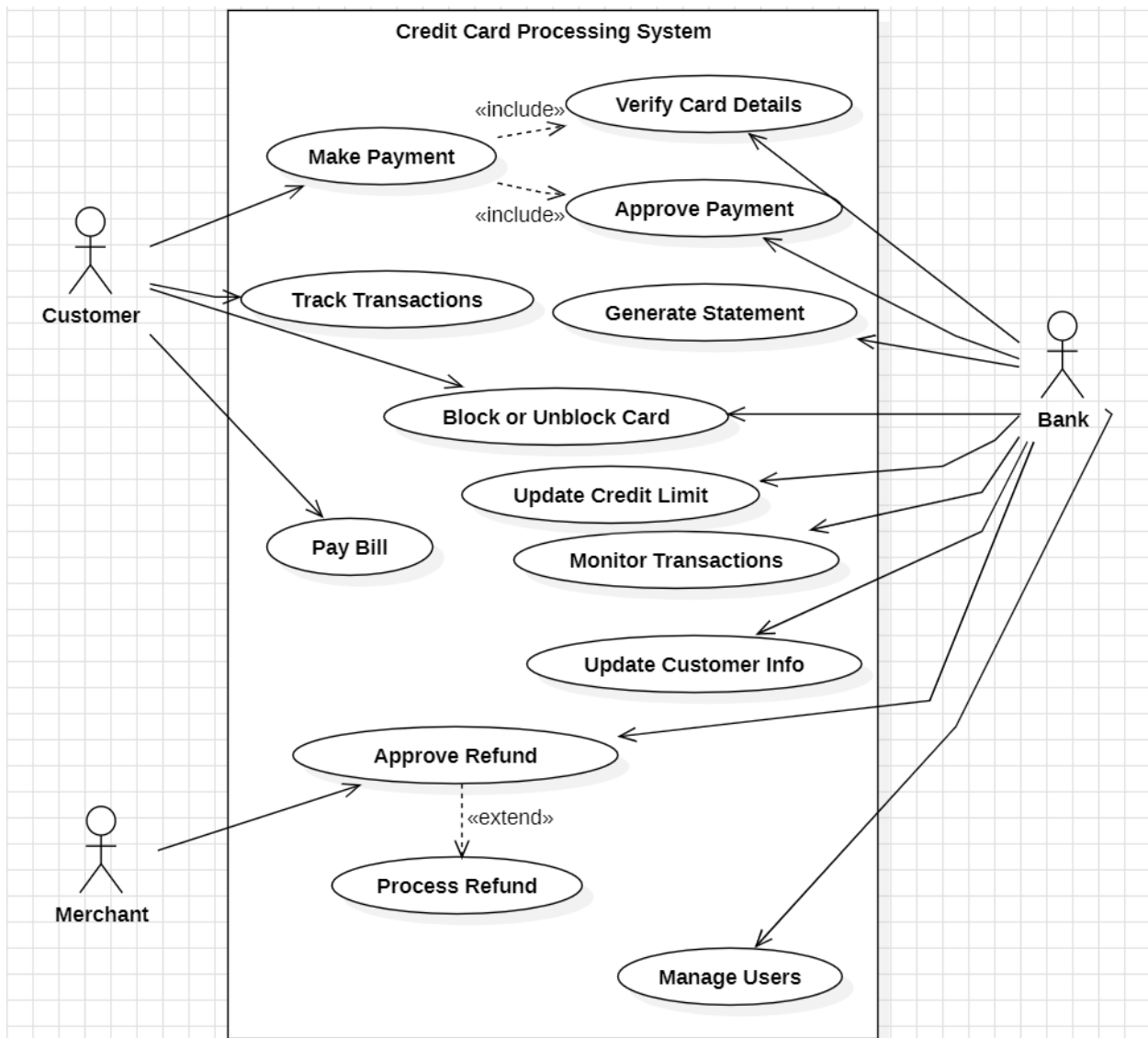


Fig 2.1 Credit Card Processing System - Use Case Diagram

The diagram depicts a Use Case Diagram for a Credit Card Processing System, highlighting the roles of the primary actors: Customer, Bank, and Merchant. The Customer interacts with the system to make payments (which includes verifying card details and approving payments), track transactions, generate statements, block or unblock cards, and pay bills. The Bank is responsible for approving payments, monitoring transactions, updating customer information, adjusting credit limits, and managing users. Additionally, merchants can request refunds, which involve approval and subsequent processing by the system. This diagram effectively illustrates the interactions and responsibilities within the credit card processing workflow.

Sequence Diagram

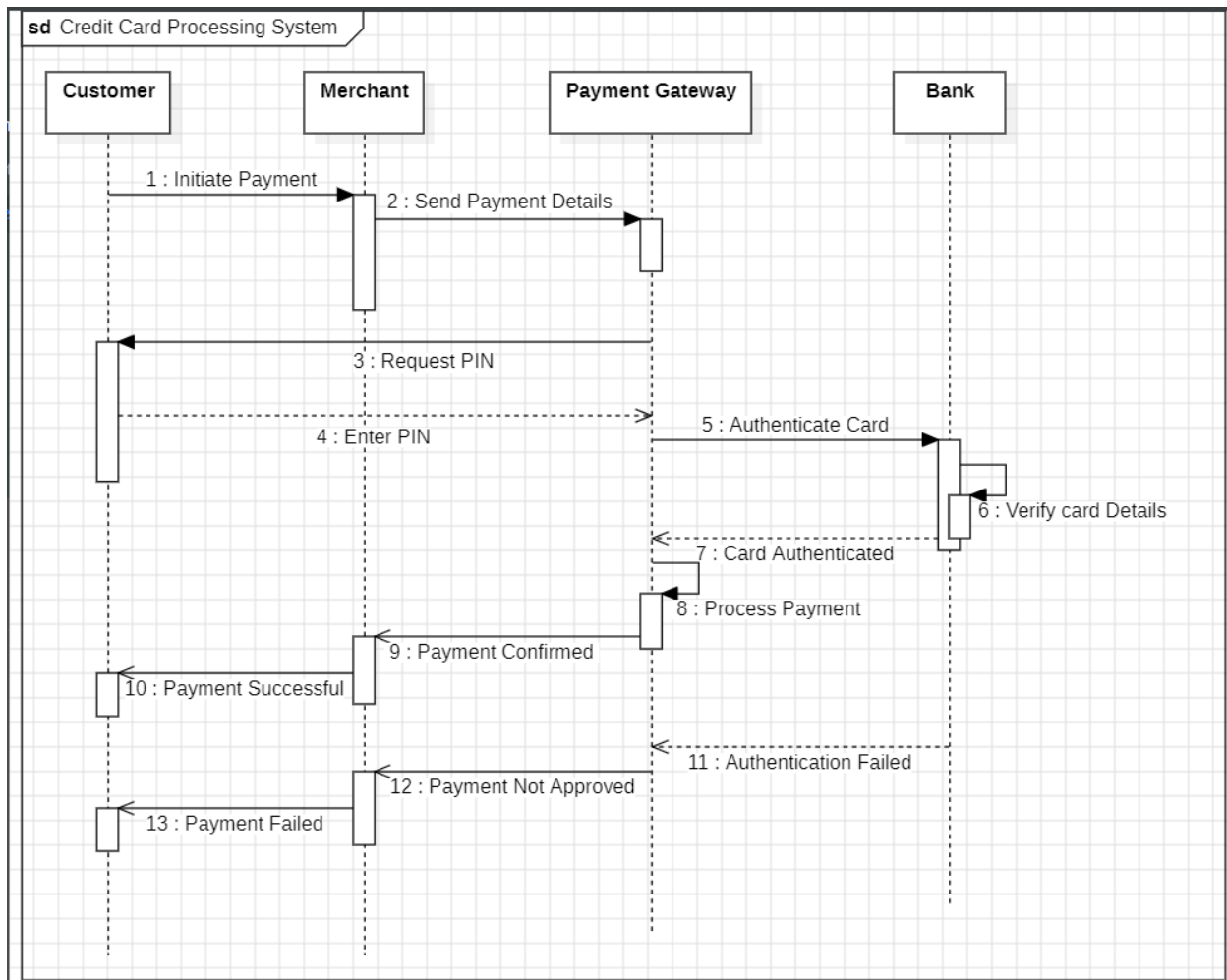


Fig 2.1 Credit Card Processing System - Sequence Diagram

The sequence diagram illustrates the process of a credit card transaction. The customer initiates the payment, and the merchant sends the payment details to the payment gateway. The payment gateway requests the customer to enter their PIN for authentication. Once the PIN is entered, the gateway authenticates the card with the bank. If the card is authenticated, the payment gateway processes the payment and confirms it to the merchant. Finally, the customer receives a notification of successful payment. If the card authentication fails, the payment is not approved

Activity Diagram

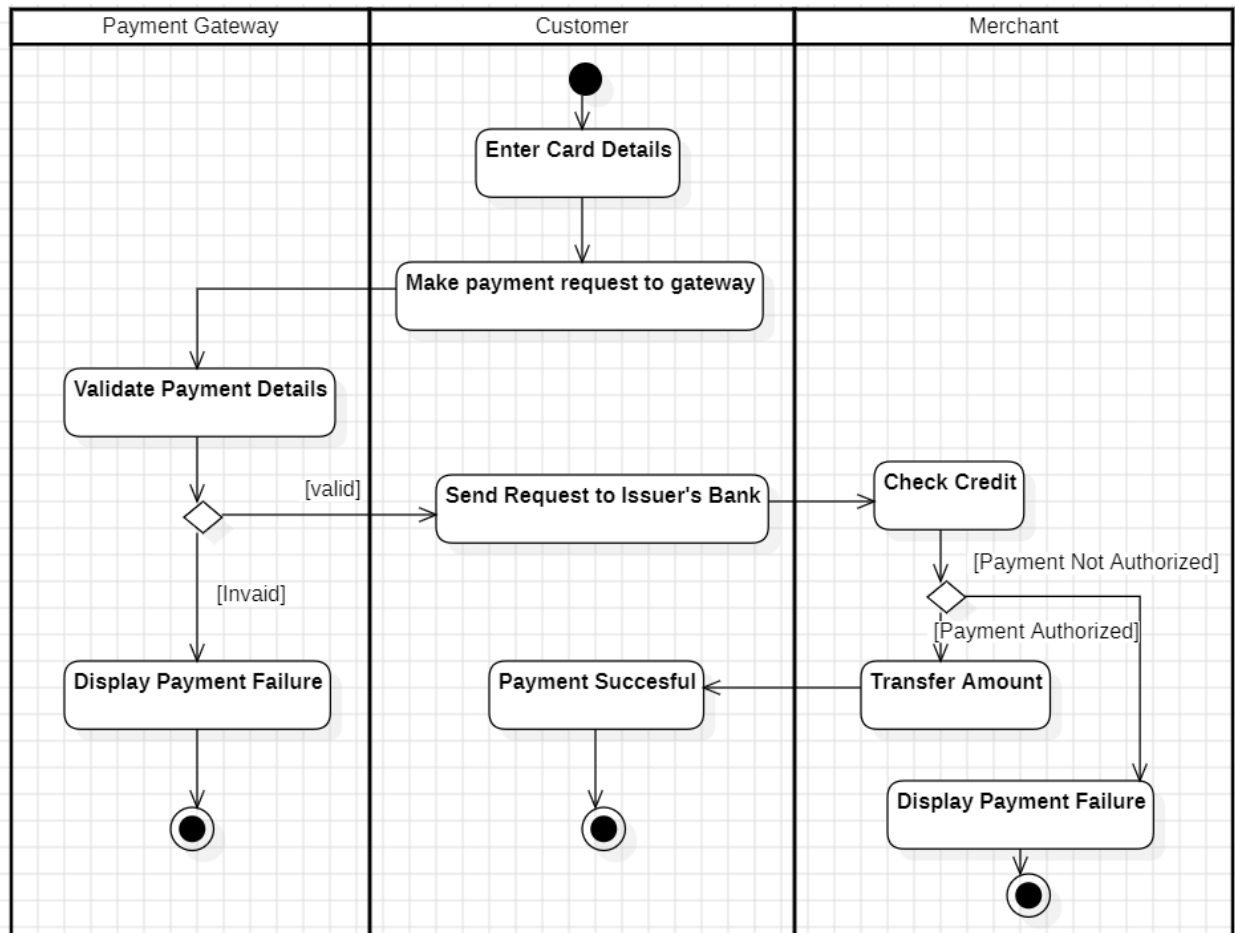
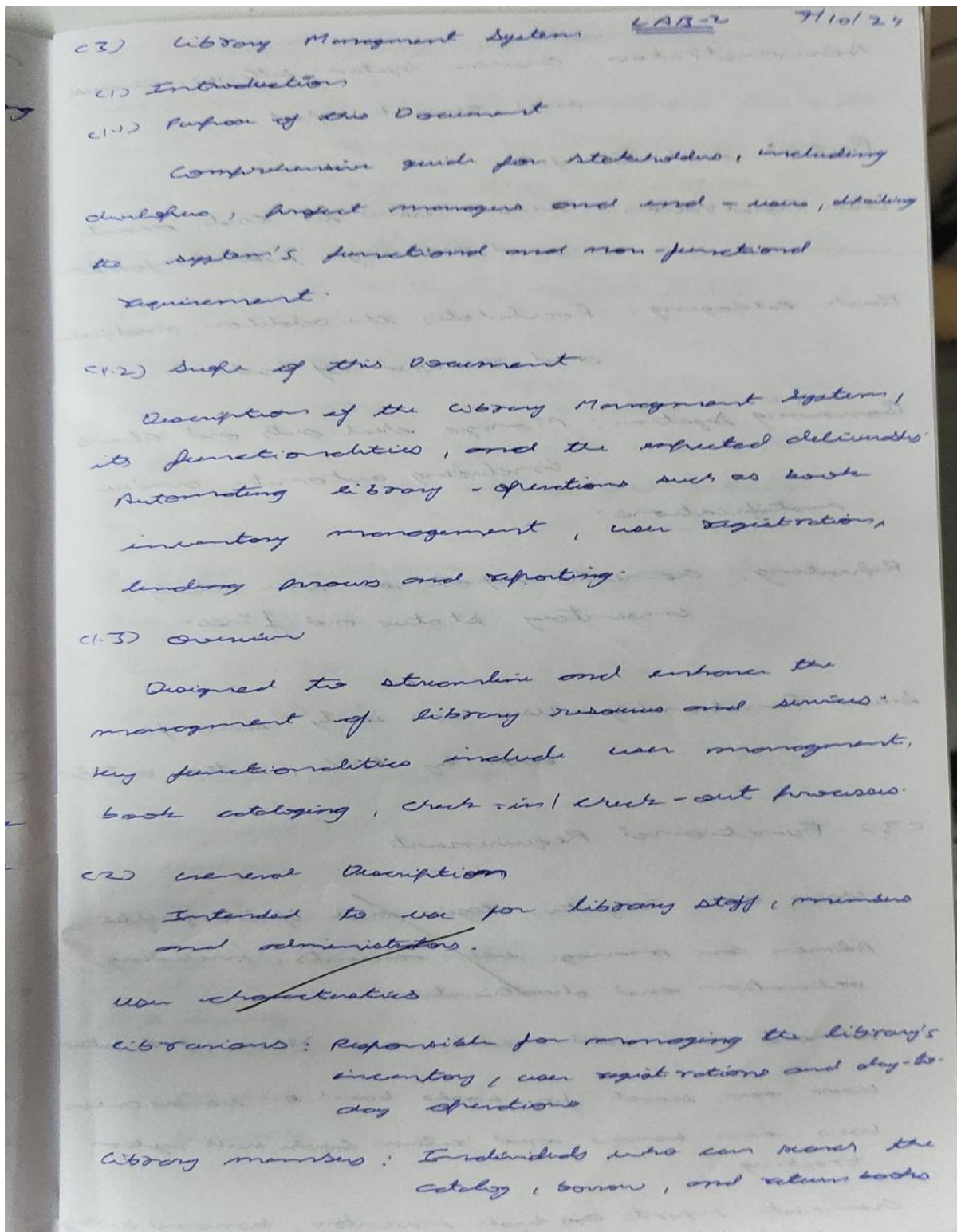


Fig 2.5 Credit Card Processing System - Activity Diagram

The activity diagram illustrates the process of a credit card transaction. The customer starts by entering their card details and making a payment request to the gateway. The gateway validates the payment details. If the details are valid, the gateway sends a request to the issuer's bank to check the credit. If the credit check is successful, the bank authorizes the payment and the gateway transfers the amount. The customer then receives a notification of successful payment. If the payment details are invalid or the credit check fails, the transaction is rejected, and the customer receives a notification of payment failure.

3. Library Management System



Software Requirement Specification

Administrators: oversee system settings, resources, and generate report.

Features and Benefits:

User Registration: Allow users to create accounts and manage personal information.

Book Cataloging: Facilitates the addition, modification, and removal of books.

Borrowing System: Manages check-outs and returns including automatic overdue notifications.

Reporting: Generate reports on user activity, inventory status and fines.

Search Functionality: Users can easily search for books by title, author or ISBN.

3. Functional Requirements

Users can register, login and update profiles.
Admins can manage user accounts, including activation and deactivation.

Admins can add, edit, and remove book entries.

Users can search for books based on various criteria.

Users can borrow and return books with system tracking.

Generate reports on book inventory, borrowed books, and fines collected.

4. Interface Requirements

Web Interface: A user-friendly interface for both staff and users, accessible through any modern web browser.

Mobile Compatibility: The system should be responsive for mobile devices.

Software Interfaces:

Database Connection: Communication with an external database for data storage and retrieval.

5. Performance Requirements

Response Time

Load Handling

Data Integrity

6. Design Constraints

Technology Stack

Hosting

Security

7. Non-functional Attributes

Security

Portability

Reliability

Scalability

Class Diagram

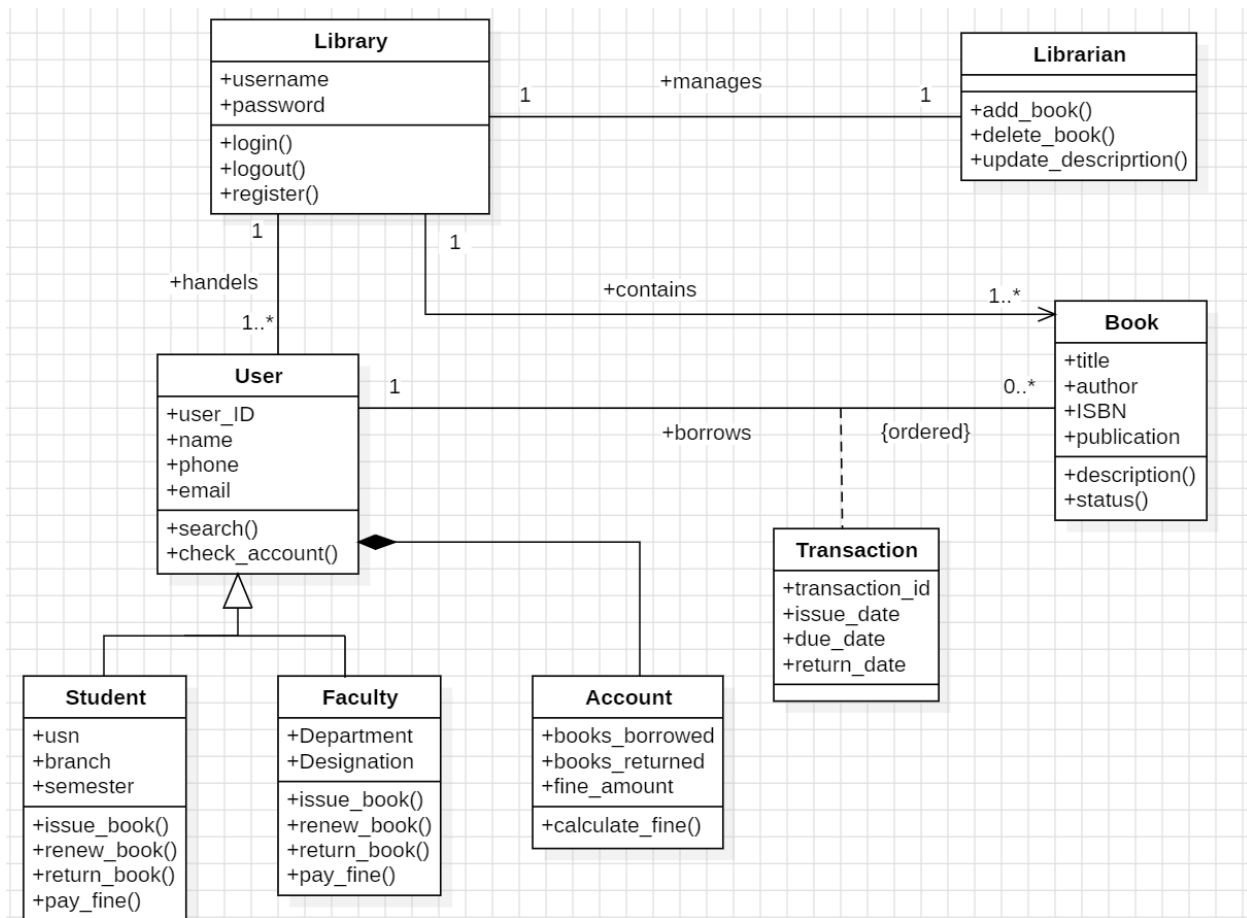


Fig 3.1 Library Management System -Class Diagram

The class diagram represents a library management system, showcasing entities like Library, Librarian, User, Book, Account, and Transaction. The Library handles the system's operations, managed by a Librarian who adds, updates, and deletes books. Users are divided into Students and Faculty, each with functionalities like issuing, renewing, and returning books, managed via their respective Accounts that track borrowed books and fines. Books store details like title, author, and status, while Transactions record borrowing and returning activities. The relationships between these entities ensure seamless management of books, users, and transactions.

State Diagram

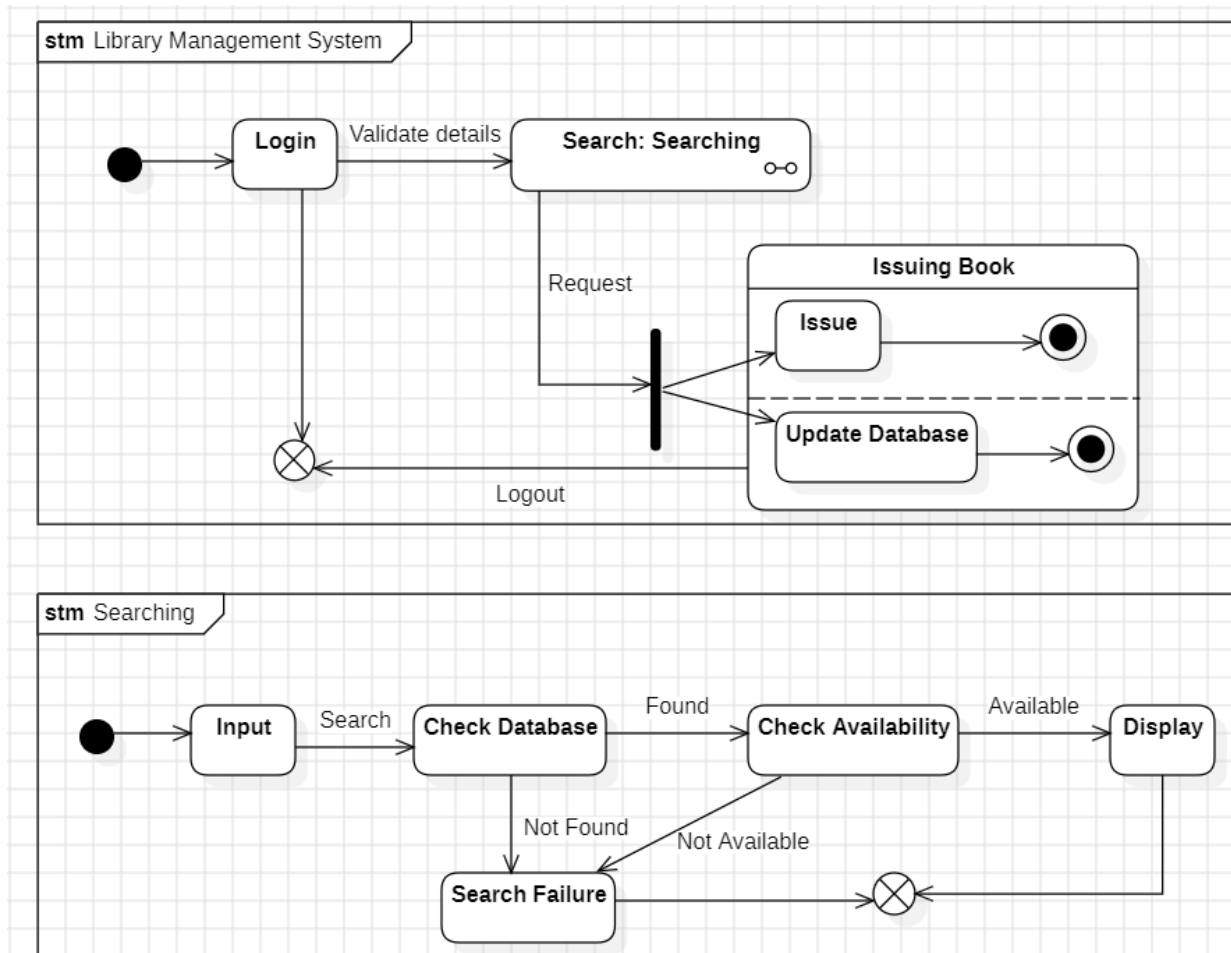


Fig 3.2 Library Management System - State Diagram

The state diagram illustrates the workflow of a library management system. It begins with user login, followed by credential validation. Upon successful login, the system enters the "Searching" state, where the user can search for books. The search process involves checking the database for matches and then checking availability. If a book is available, its details are displayed to the user, who can then request to issue it. The system updates its database accordingly. If the search yields no results or the book is unavailable, the system transitions to the "Search Failure" state. At any point, the user can log out of the system.

Use Case Diagram

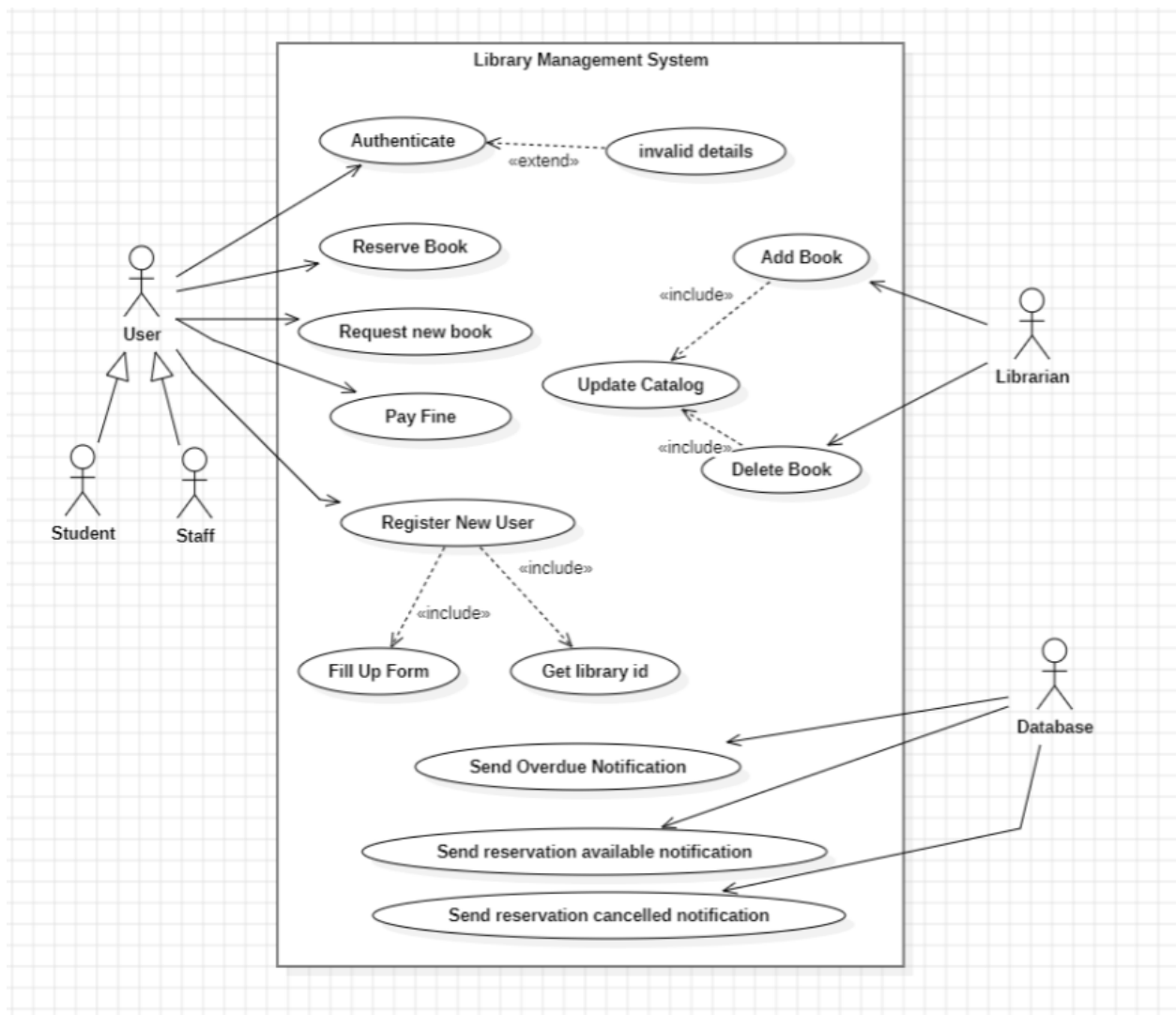


Fig 3.3 Library Management System - Use Case Diagram

The Library Management System is designed to manage the library's resources and user interactions. The system has three main actors: User, Librarian, and Database. The User can reserve books, request new books, pay fines, and register as a new user. The Librarian can add books to the catalog, update the catalog, delete books, and send overdue notifications. The Database stores and manages all the information related to the library, users, and books. The system includes use cases for authentication, filling up forms, and getting library IDs, which are further elaborated by the "include" relationships. This system aims to streamline library operations and provide a convenient experience for users.

Sequence Diagram

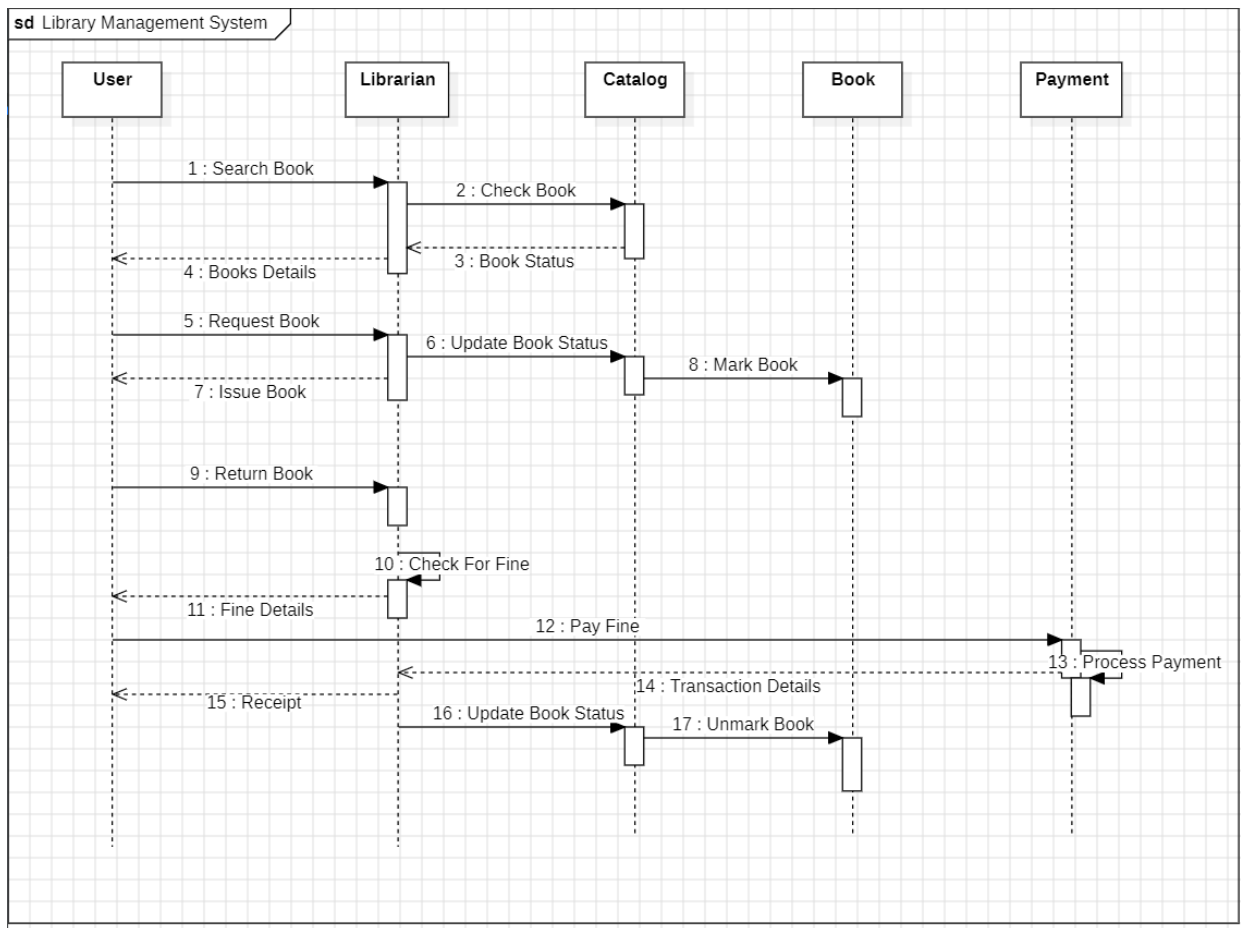


Fig 3.4 Library Management System - Sequence Diagram

The sequence diagram illustrates the process of a user borrowing a book from the library. The user begins by searching for a book in the library catalog. The catalog then searches for the book and returns the results to the user. The user then requests to borrow the book, and the library system checks its availability. If the book is available, the system issues the book to the user and updates its records. Finally, the user receives a receipt confirming the checkout. This diagram highlights the automated steps involved in the process and the interactions between the user and the library system.

Activity Diagram

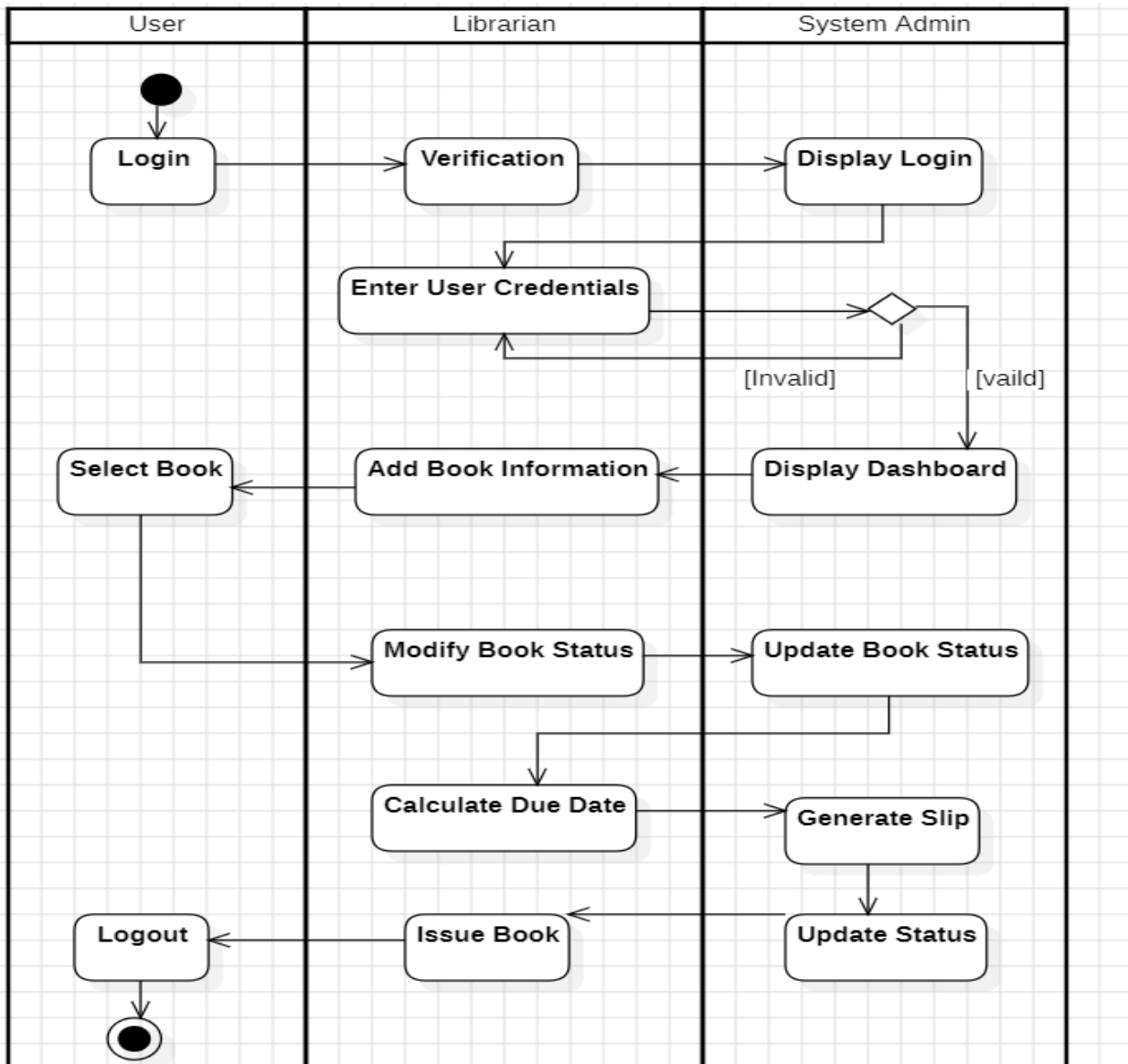


Fig 3.5 Library Management System - Activity Diagram

The activity diagram outlines the workflow of a library management system. It starts with a user logging in, followed by credential verification. Successful login grants access to book selection for the user. Simultaneously, librarians can add new books or modify existing book information. System administrators possess the authority to update book statuses and generate slips related to book transactions. Upon book selection, the system calculates the due date and issues the book to the user, updating the database accordingly.

4. Stock Maintenance System

(4) Stock Maintenance System

1. Introduction

1.1 Purpose of this Document

To outline the requirements for the stock maintenance system. Comprehensive guide for stakeholders, developers and testers, ensuring a clear understanding of the system's functionalities and constraints.

1.2 Scope of this Document

Covers the overall design, functionality and requirement of the Stock Maintenance System, which is intended to assist businesses in managing their inventory efficiently.

1.3 Overview

Designed to facilitate the tracking and management of stock in warehouses or retail environments. It will offer features such as stock tracking, order management, supplier management and reporting tools.

(2) General Description

The Stock Maintenance System will provide users, primarily inventory managers and staff, with an intuitive interface for managing stock levels. Key features include:

3) Functional Requirements →

1) Stock Level Tracking → The system should allow users to input and track current stock levels for each item in inventory.

2) Orders and Purchases Recording → Users must be able to record orders and purchases, and the system should automatically update the stock levels.

3) Low Stock Alerts → The system will automatically notify users when stock falls below a predefined threshold.

4) Reporting → The system must generate reports on stock levels, stock history, sales, purchases and more.

5) Data Import/Export → Users must be able to import and export stock data via CSV or Excel files.

4) Interface Requirements →

1) User Interface → A web-based graphical interface where users can interact with the system, input data and generate reports.

2) API Integration → The system should integrate with external orders and purchasing systems via APIs to ensure real-time stock updates.

5) Performance Requirements →

- 1) Response Time
- 2) Availability
- 3) Scalability
- 4) Data Processing

6) Design Constraints →

- 1) Technology Stack

2) Hardware

3) Database

4) Data Security

7) Non-functional Requirements →

- 1) Security
- 2) Portability
- 3) Reliability
- 4) Scalability
- 5) Usability
- 6) Data Integrity

8) Preliminary Schedule and Budget:

- Estimated Budget: £50,000
- Development Timeline: 6 months

Class Diagram

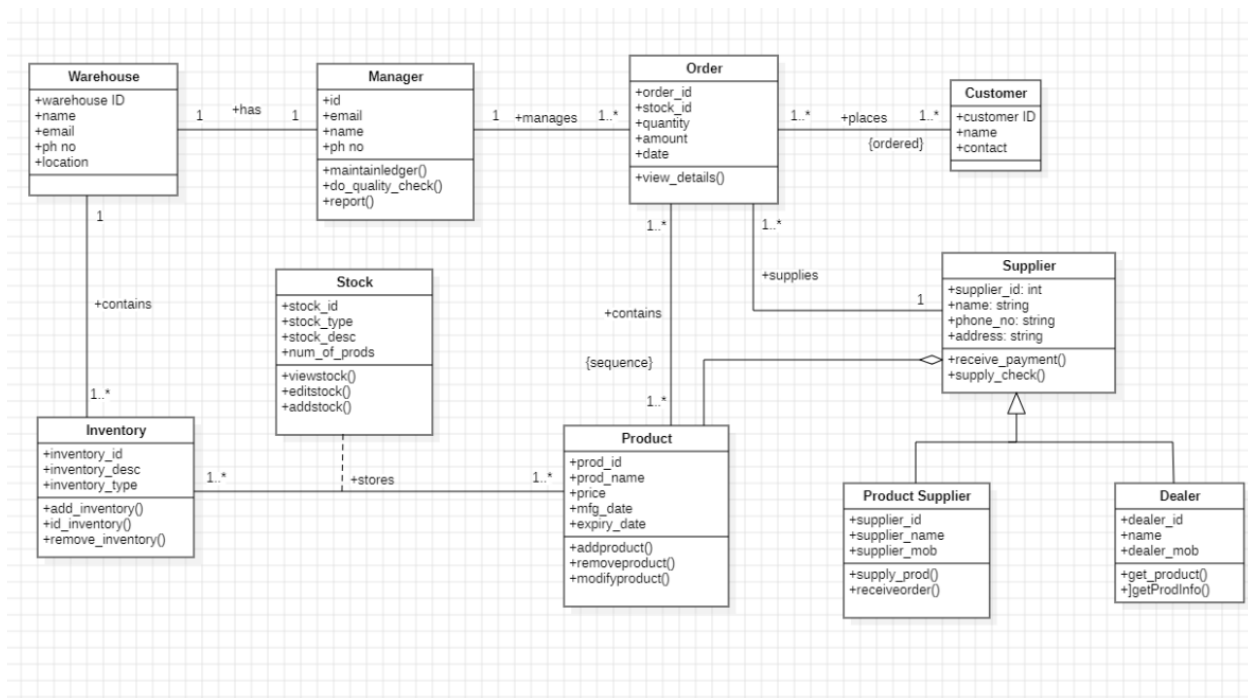


Fig 4.1 Stock Maintenance System - Class Diagram

The class diagram illustrates a warehouse inventory and order management system. The Warehouse contains multiple Inventory items, managed by a Manager who oversees operations like reporting and quality checks. Stock stores details about products, which are managed with functionalities like addition and modification. Orders placed by Customers link products to quantities and amounts, while Suppliers, including Product Suppliers and Dealers, handle the supply of products to the warehouse. The diagram highlights the interactions between inventory, stock, orders, and suppliers within the system.

State Diagram

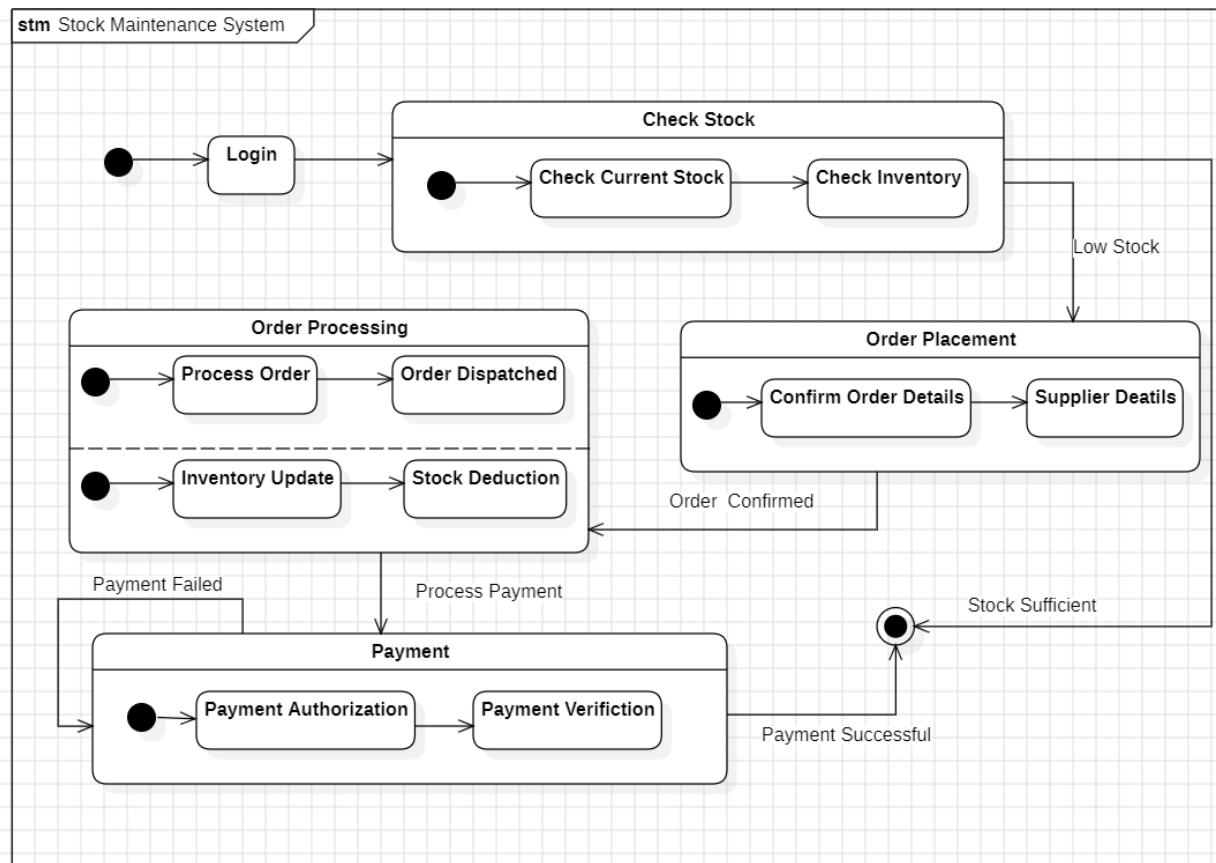


Fig 4.2 Stock Maintenance System - State Diagram

The state diagram illustrates the stock maintenance system's workflow. It starts with a user logging in. The system then checks current stock and inventory levels. If stock is low, the system transitions to the "Order Placement" state, where order details are confirmed and supplier details are obtained. After the order is confirmed, the system moves to the "Order Processing" state, where the order is processed and dispatched. During order processing, stock is deducted and inventory is updated. Finally, the system transitions to the "Payment" state, where payment is authorized and verified. Upon successful payment, the system returns to the "Check Stock" state to monitor inventory levels.

Use Case Diagram

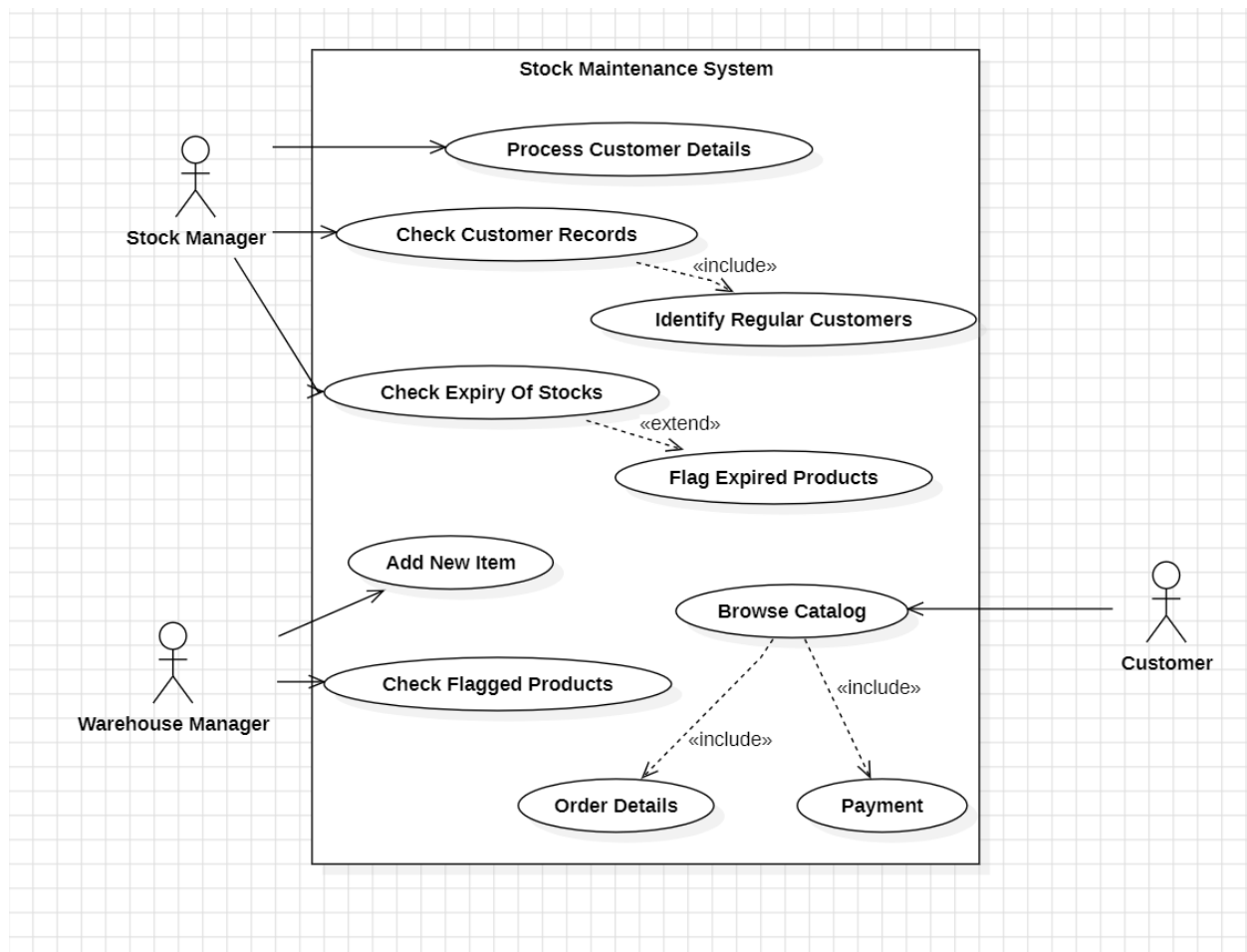


Fig 4.3 Stock Maintenance System - Use Case Diagram

The Stock Maintenance System is designed to manage inventory and customer interactions for a business. The system has three main actors: Stock Manager, Warehouse Manager, and Customer. The Stock Manager can process customer details, check customer records, and identify regular customers. They can also check the expiry of stocks and flag expired products. The Warehouse Manager can add new items to the inventory and check flagged products. The Customer can browse the catalog, place orders, and make payments. The system includes use cases for order details and payment, which are further elaborated by the "include" relationships. This system aims to streamline inventory management and provide a seamless experience for customers.

Sequence Diagram

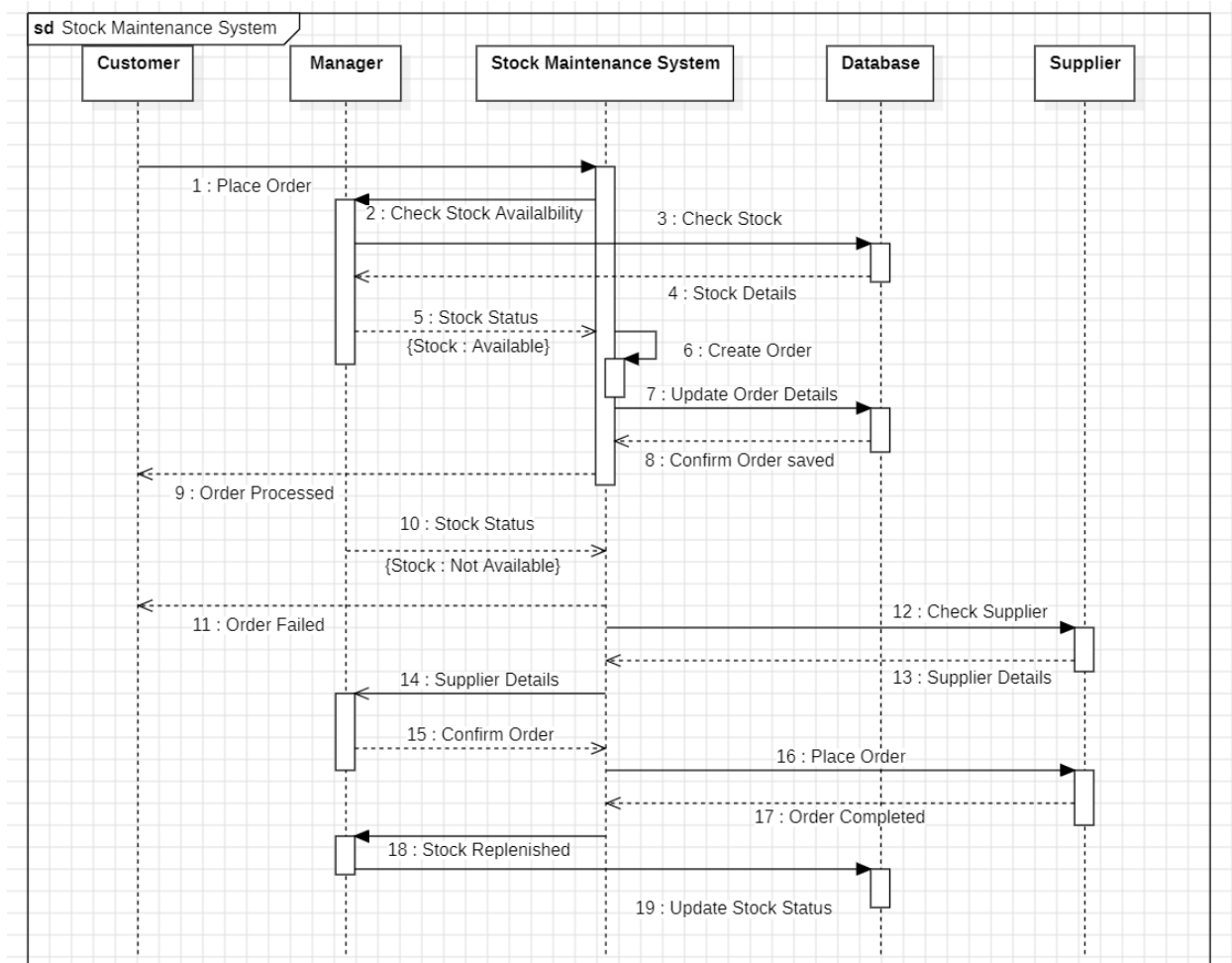


Fig 4.4 Stock Maintenance System - Sequence Diagram

The sequence diagram outlines the order fulfillment process in a stock maintenance system. It begins with the customer placing an order. The manager then checks stock availability, and the system verifies stock levels in the database. If stock is sufficient, the order is created and processed. If stock is insufficient, the system checks with suppliers, places orders, and updates stock levels once the replenishment is complete. The system communicates order status updates to the customer throughout the process. This diagram illustrates the interactions between the customer, manager, database, and suppliers, highlighting the steps involved in fulfilling an order effectively.

Activity Diagram

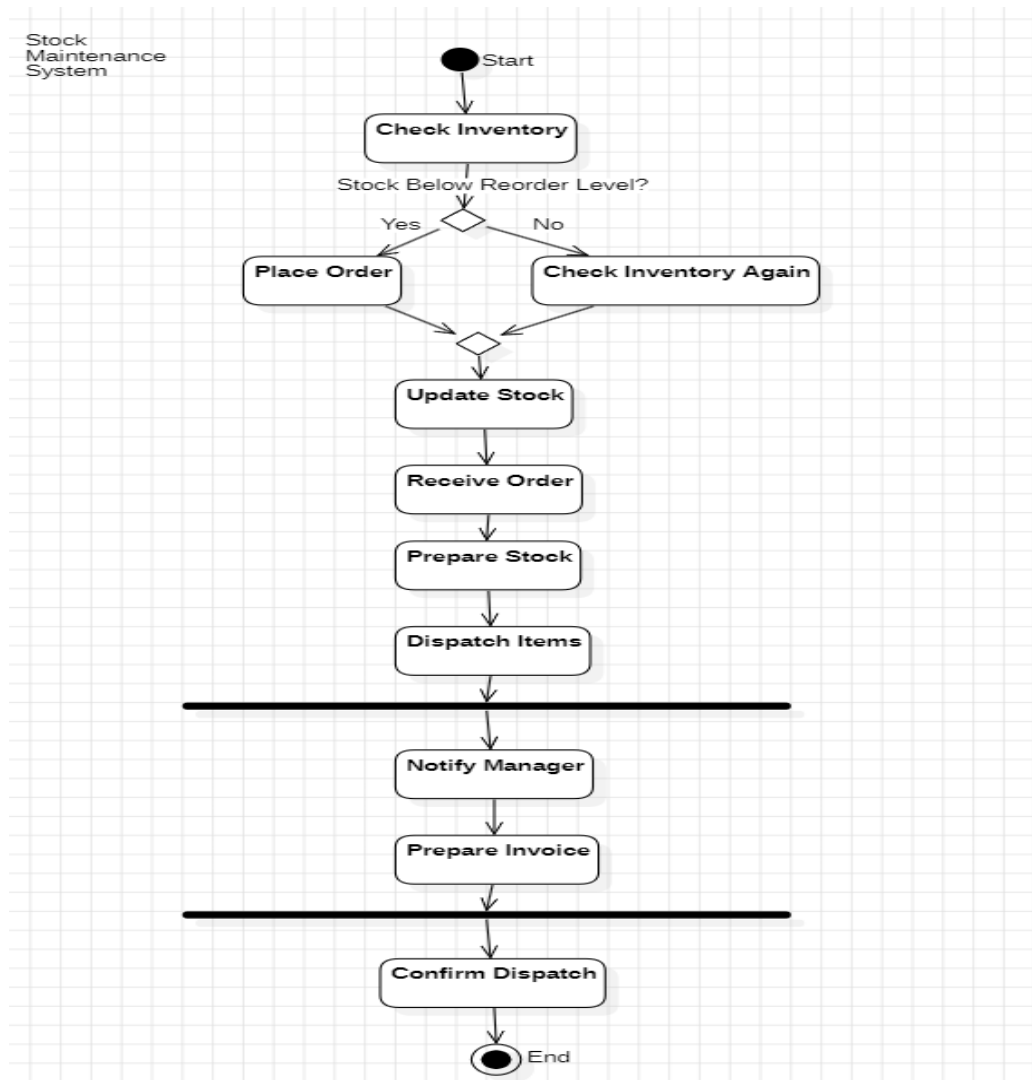


Fig 4.5 Stock Maintenance System-Activity Diagram

This activity diagram represents the workflow of a stock maintenance system. It begins with checking inventory levels, followed by a decision point to assess if stock is below the reorder level. If stock is low, an order is placed, otherwise, inventory is rechecked. Upon receiving the order, the stock is updated, prepared, and dispatched. The process then involves notifying the manager, preparing the invoice, and confirming the dispatch, concluding the workflow.

5. Passport Automation System

5.1 Passport Automation System →

2. Introduction:

2.1 Purpose of this Document

outlines the specifications for automating the passport application, processing and issuance workflow. It is intended to ensure that all stakeholders, including developers, testers and government officials are aligned on the system's objectives and functionalities.

2.2 Scope of this Document:

This defines the specifications for the development of a passport automation system, which will enable citizens to apply for a passport online, track the application status and receive notifications about the application.

2.3 Overview:

The Passport Automation System is designed to provide a fully digital platform for passport applications and processing. Users can submit their applications online, book appointments for document verification and track the application progress in real-time.

5.2 Functional Requirements →

2.1 User Registration and Authentication: The system must allow users to create accounts and login securely using credentials.

Software Requirement Specification

(b) Online Passport Application \rightarrow Users should be able to fill out the passport application form online and submit all required documents digitally.

(c) Document Verification \rightarrow The system will automatically verify uploaded documents using predefined rules and integrate with national databases for validation.

(d) Appointment Scheduling \rightarrow Users can schedule an appointment for biometric data collection at a passport office.

(e) Real-Time Application Status \rightarrow Users will receive real-time updates on their application status, including stages such as verification, police clearance.

(4) Interface Requirement \rightarrow

(a) User Interface

(b) API Integration

(c) Biometric Interface

(d) Database

(5) Performance Requirements \rightarrow

(a) Response Time

(b) Scalability

(c) Data Processing

(d) Availability

(e) Data Throughput

Class Diagram

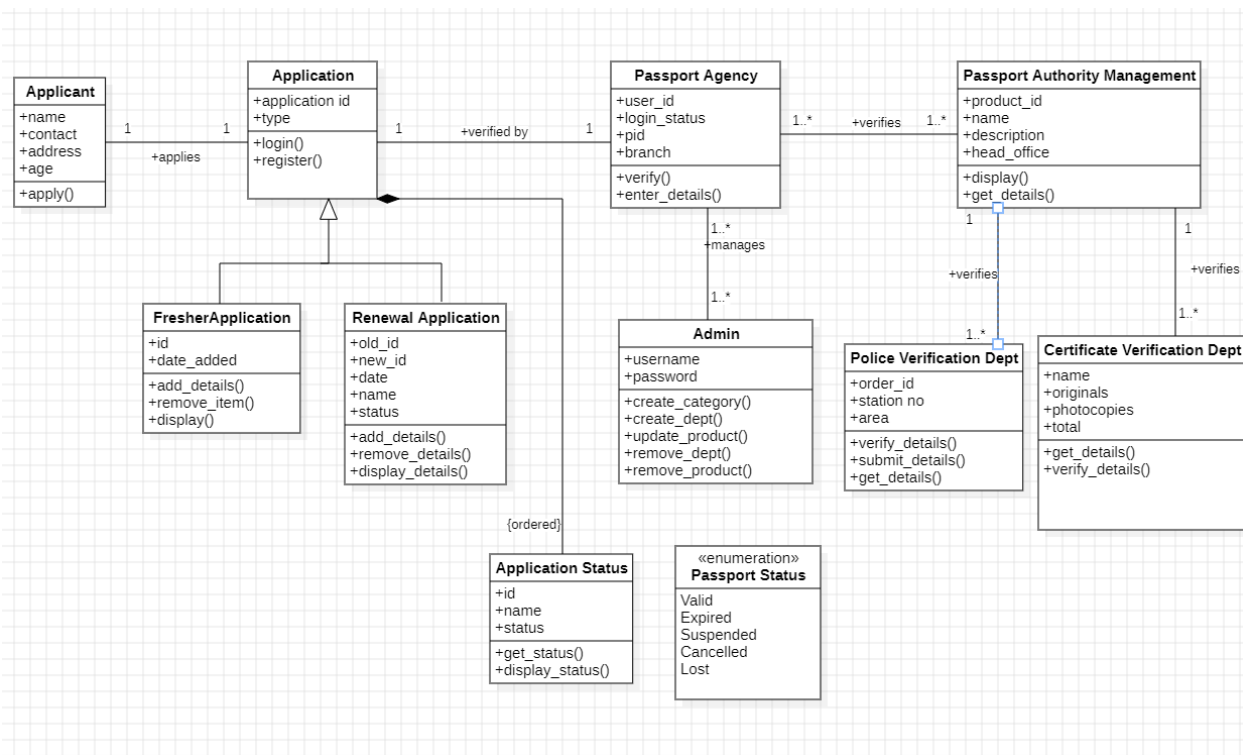


Fig 5.1 Passport Automation System - Class Diagram

The class diagram depicts the structure of a passport application and verification system. It illustrates various entities, such as Applicant, Application, and its specialized forms: FresherApplication and RenewalApplication. The Application class is associated with Applicant, who can apply and register for passport services. The system includes a Passport Agency and its management under Passport Authority Management, which oversees verification processes through departments like Police Verification and Certificate Verification. The diagram also involves an Admin class responsible for managing categories, departments, and products. Key features include status tracking through Application Status and Passport Status enumeration. Relationships between classes are depicted with multiplicity, inheritance, and composition, highlighting functionalities like verifying details, managing applications, and updating statuses.

State Diagram

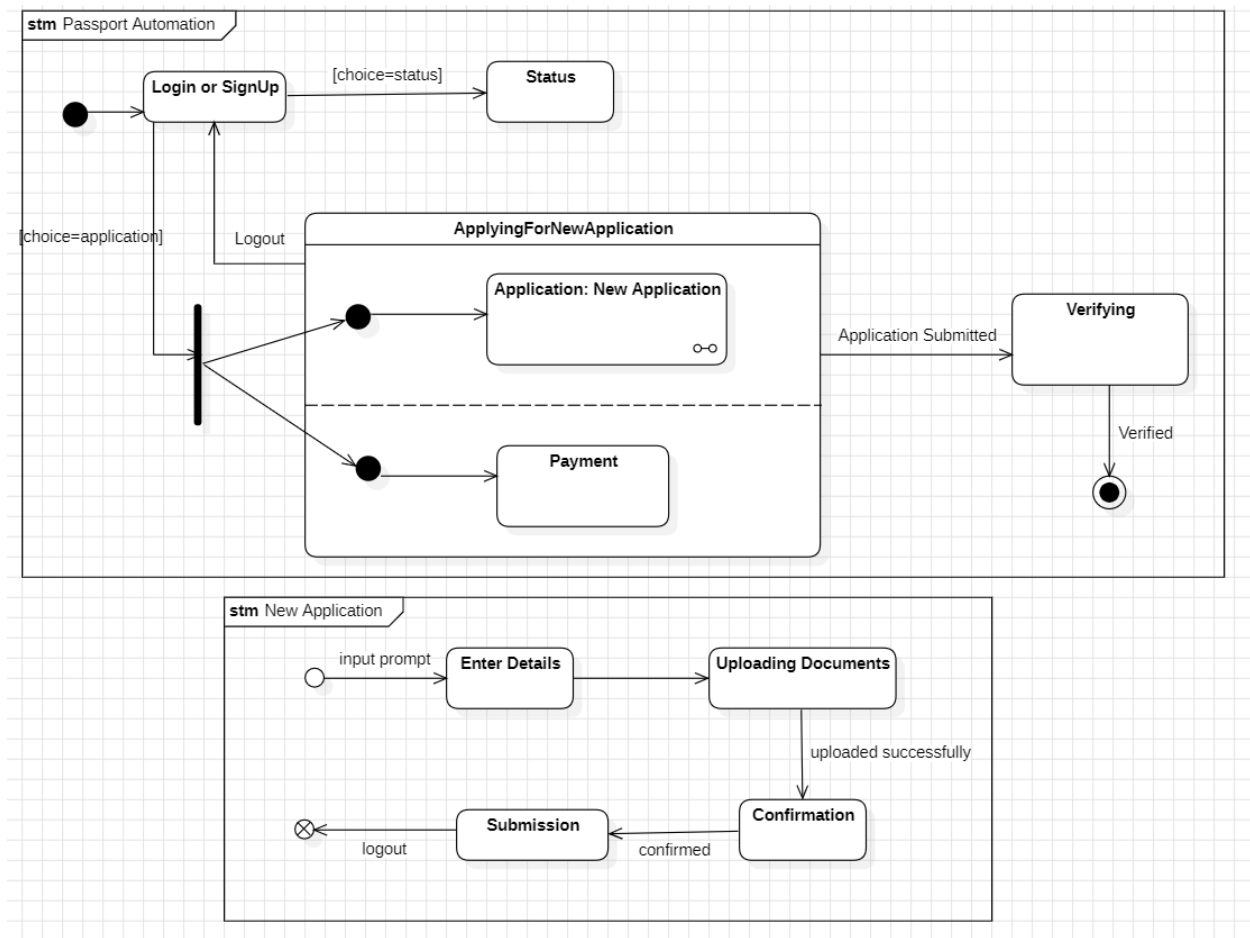


Fig 5.2 Passport Automation System - State Diagram

The state diagram illustrates the passport automation system. The system starts with the user logging in or signing up. After login, the user can choose to check the status of their application or apply for a new one. If the user chooses to apply, they enter the "ApplyingForNewApplication" state. Within this state, the user fills out the application form, uploads documents, and submits the application. Once submitted, the application enters the "Verifying" state. If the application is verified successfully, the user receives a confirmation. The user can also log out at any point during the process.

Use Case Diagram

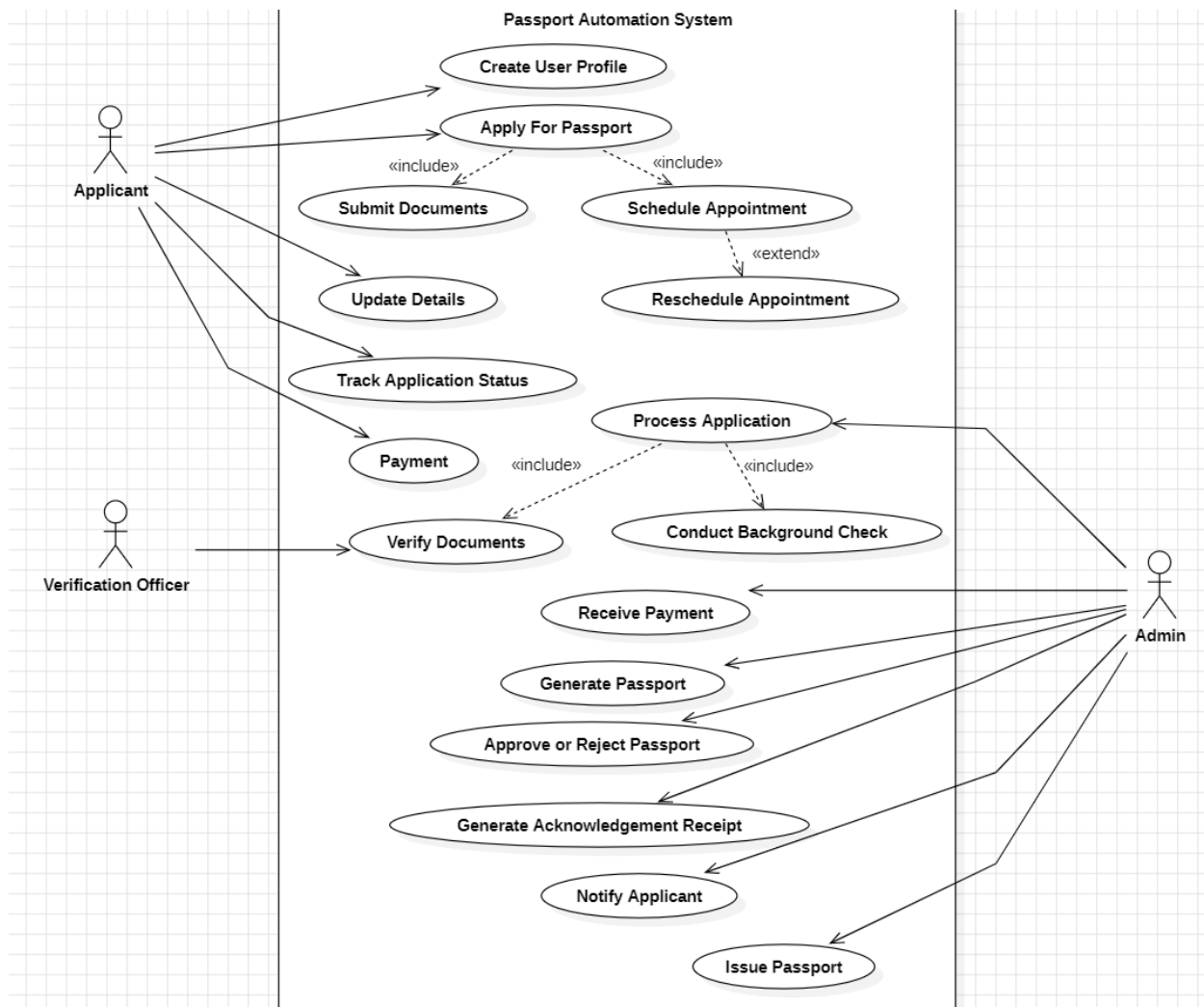


Fig 5.3 Passport Automation System - Use Case Diagram

The diagram illustrates a Use Case Diagram for a Passport Automation System, outlining the interactions between the system and its primary actors: Applicant, Verification Officer, and Admin. The Applicant begins by creating a user profile and applying for a passport, which includes submitting documents, scheduling (or rescheduling) appointments, making payments, and tracking application status. The Verification Officer is responsible for verifying documents and supporting the application processing. The Admin plays a key role in processing applications, conducting background checks, receiving payments, approving or rejecting passport requests, and issuing passports. Additional use cases include generating acknowledgments and notifying applicants of the application's status. This diagram effectively demonstrates the workflow and responsibilities of each actor in the passport issuance process.

Sequence Diagram

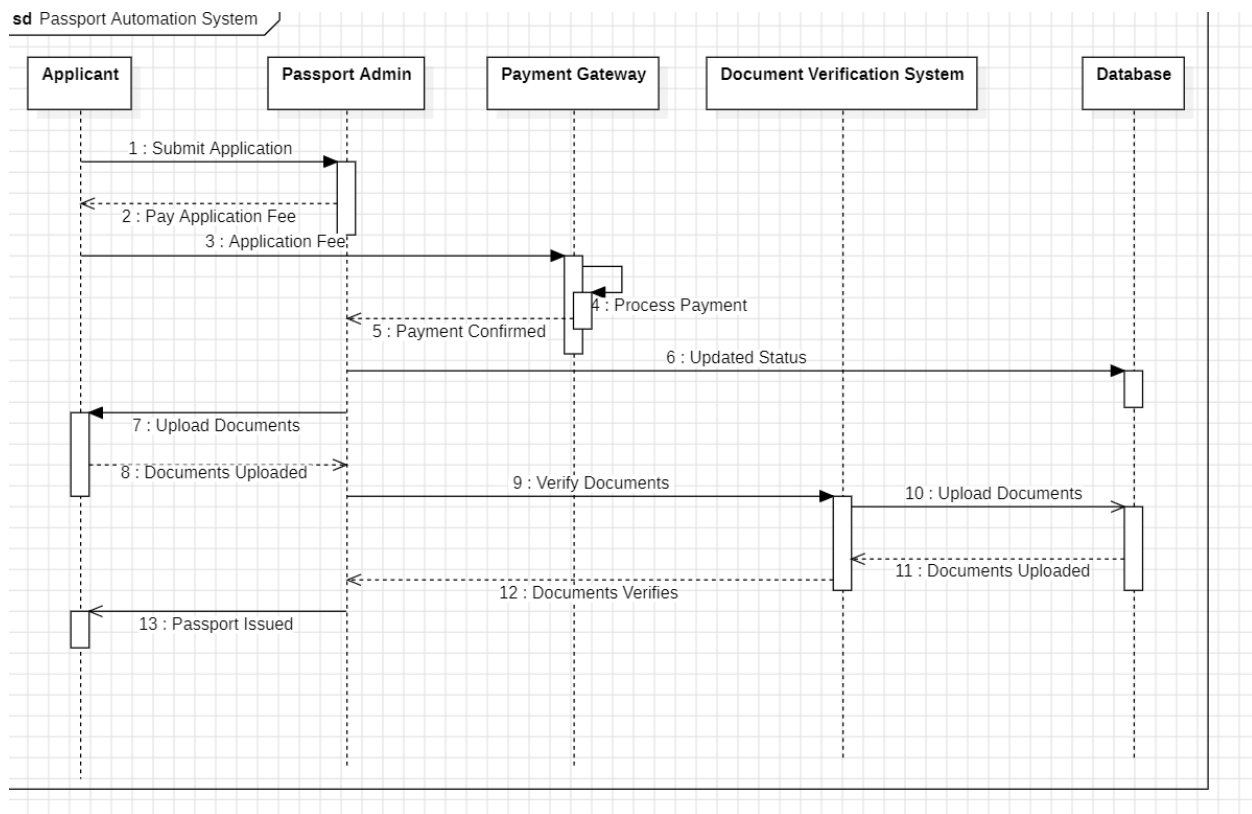


Fig 5.4 Passport Automation System - Sequence Diagram

The sequence diagram illustrates the process of applying for a passport. The applicant starts by submitting an application and then pays the application fee. The payment gateway processes the payment and updates the status. The applicant then uploads the required documents, which are verified by the document verification system. Once the documents are verified, the passport is issued to the applicant. This diagram shows the interactions between the applicant, passport admin, payment gateway, document verification system, and database throughout the passport application process.

Activity Diagram

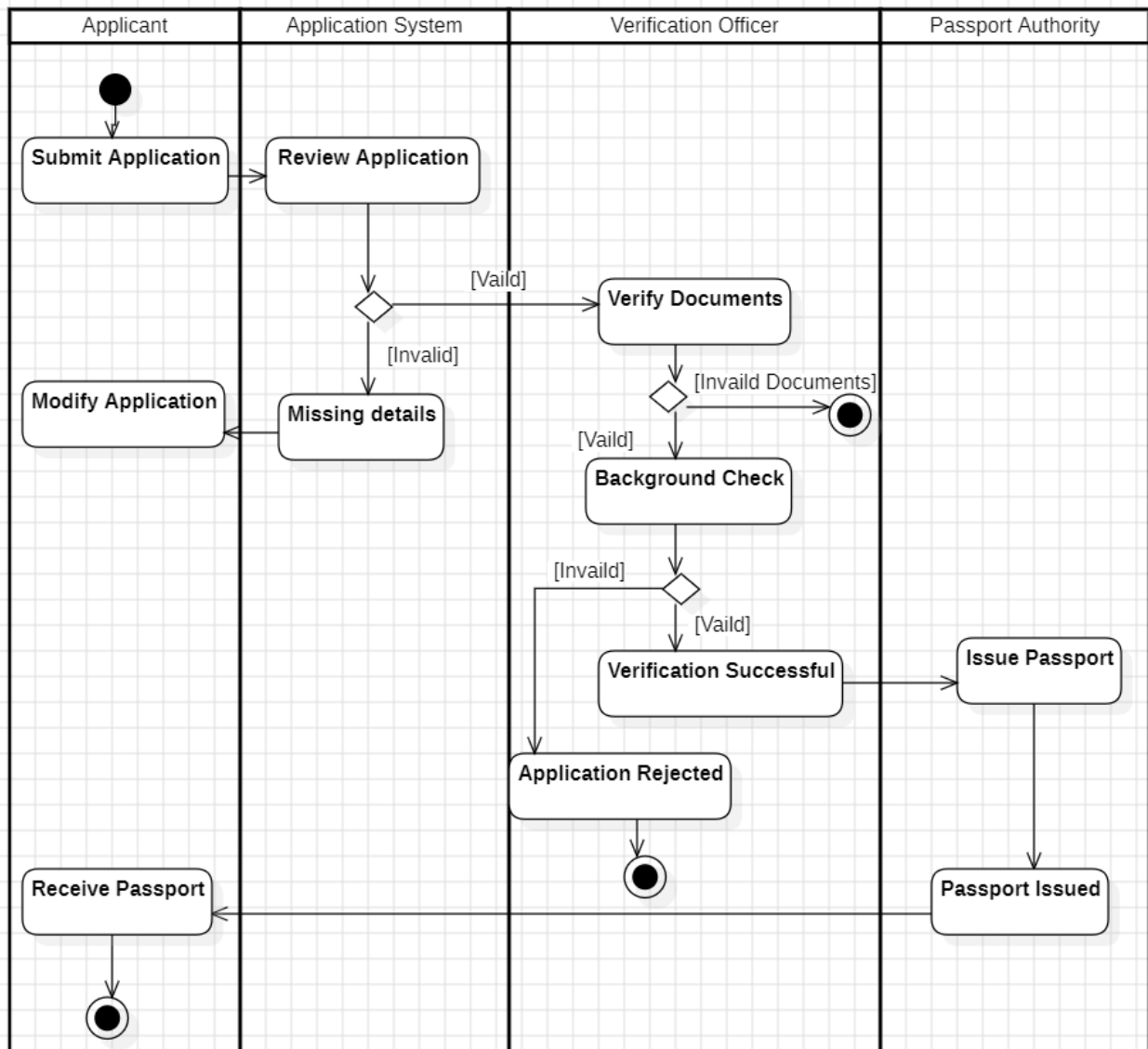


Fig 5.5 Passport Automation System - Activity Diagram

The activity diagram illustrates the passport application process. It starts with the applicant submitting an application. The application system reviews the application. If the application is complete, it proceeds to document verification. If invalid documents are found, the application is rejected. If valid, a background check is conducted. If the background check is clear, the verification is successful, and the passport is issued. If any stage fails, the application is rejected. The applicant can receive the passport once it's issued.