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

Lab Record

Software Engineering and Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SIRIGIREDDY PRANAV REDDY

1BM22CS281

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 Analysis and Design(22CS6PCSEO) laboratory has been carried out by **SIRIGIREDDY PRANAV REDDY(**1BM22CS281) during the 5th Semester Oct24-Jan2025.

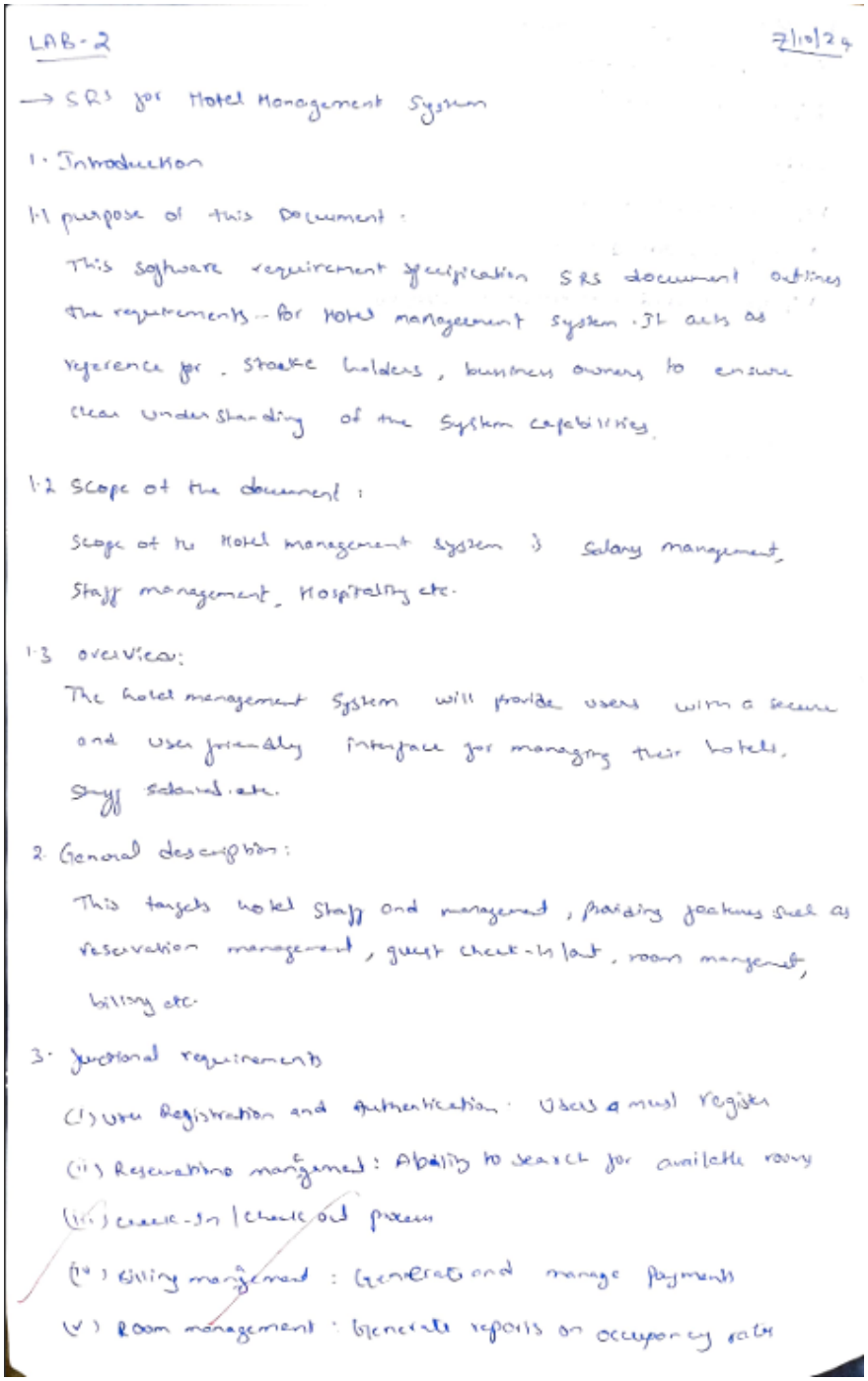
Signature of the Faculty Incharge:

Prameetha Pai
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

1. Hotel Management System

Software Requirement Specification



LAB-2 7/10/24

→ SRS for Hotel Management System

1. Introduction

1.1 purpose of this Document :

This software requirement specification SRS document outlines the requirements for Hotel management system. It acts as reference for stake holders, business owners to ensure clear understanding of the system capabilities.

1.2 Scope of the document :

Scope of the Hotel management system : Salary management, Staff management, Hospitality etc.

1.3 overview:

The hotel management system will provide users with a secure and user friendly interface for managing their hotels, staff salaries etc.

2. General description :

This targets hotel staff and management, providing features such as reservation management, guest check-in/out, room management, billing etc.

3. Functional requirements

(i) User Registration and Authentication : Users must register

(ii) Reservation management : Ability to search for available rooms

(iii) check-in/check out process

(iv) Billing management : Generate and manage payments

(v) Room management : Generate reports on occupancy rate

4. Interface Requirements:

(i) User Interface (UI)

(ii) API's

(iii) Database Interface

5. Performance Requirements

(i) Response Time.

(ii) Concurrent users

(iii) Data Storage

6. Design constraints

(i) Platform

(ii) Database

(iii) Framework

7. Non-functional Attributes

(i) Security

(ii) Scalability

(iii) Reliability

8. Preliminary schedule and Budget

(i) Estimated Development Duration

(ii) Estimated Budget

Class Diagram

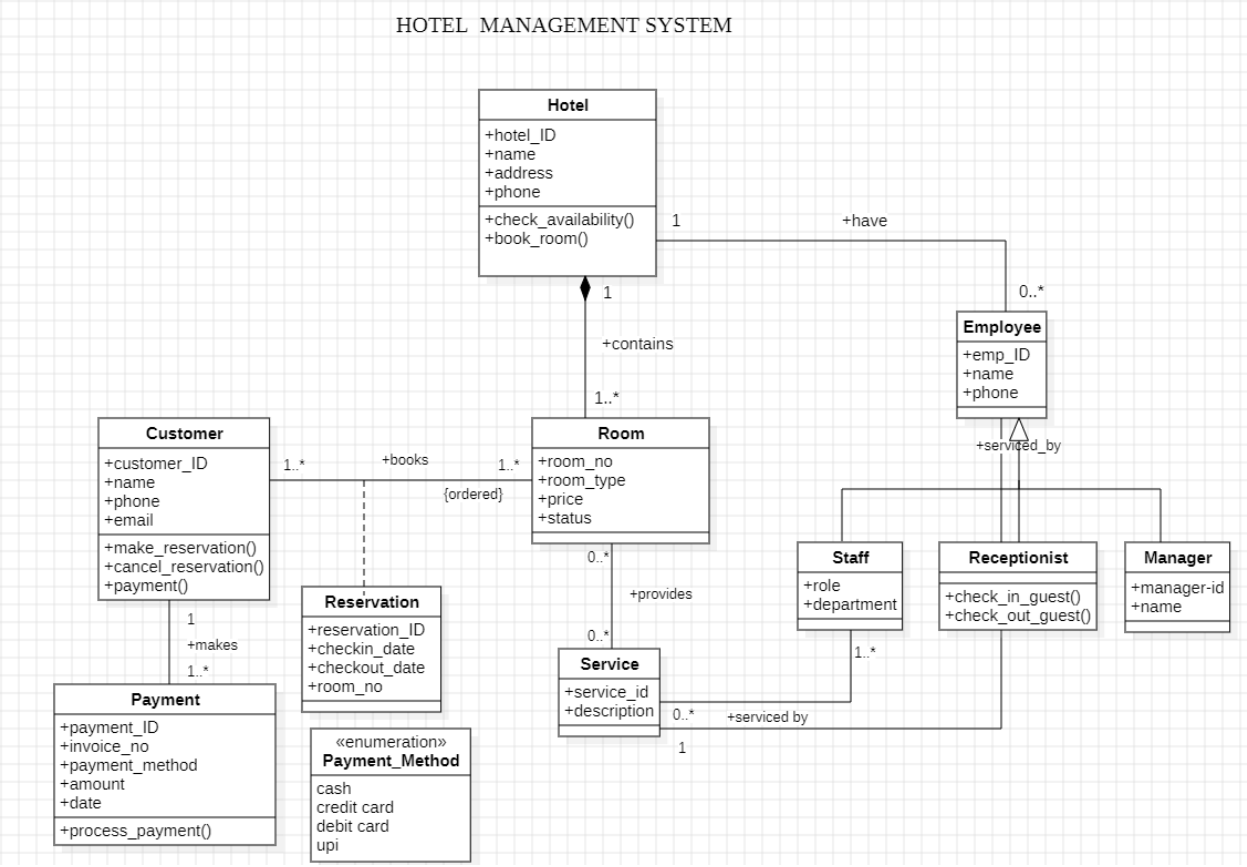


Fig1.1 Hotel Management System - Class Diagram

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

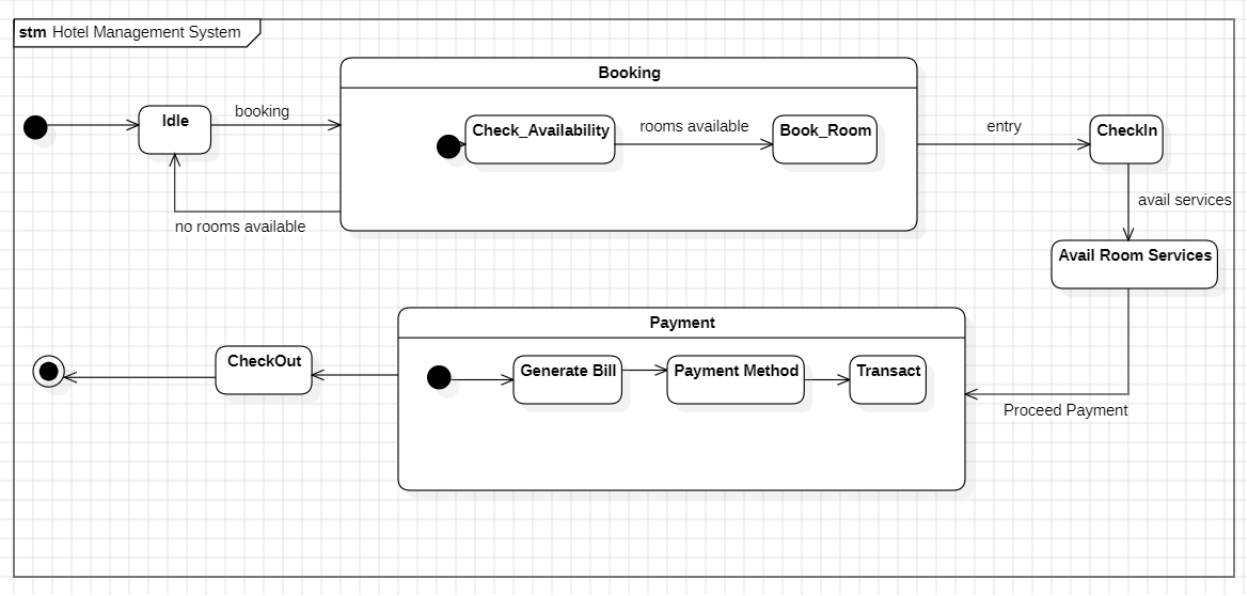


Fig1.2 Hotel Management System - State Diagram

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

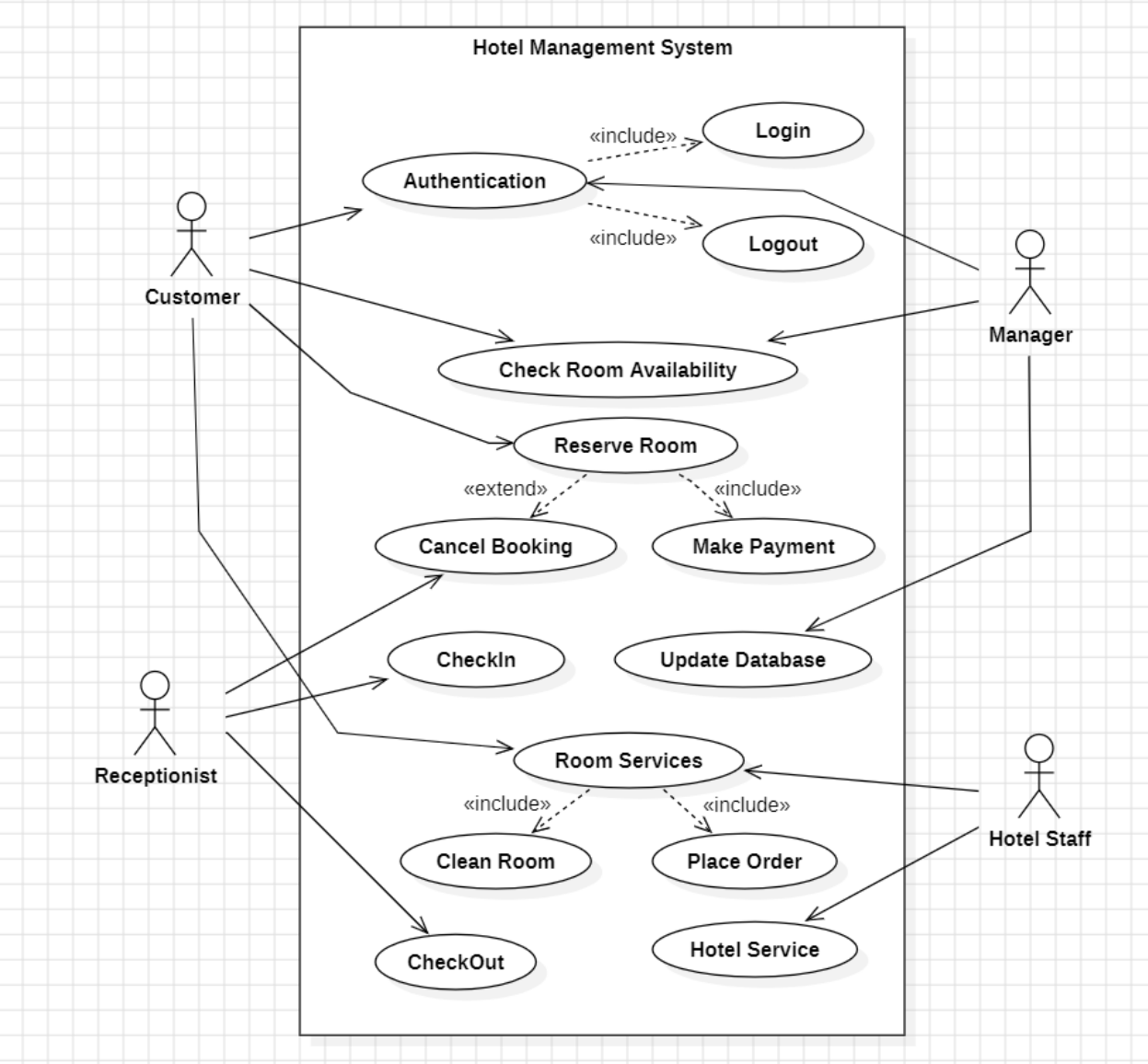


Fig1.3 Hotel Management System - Use Case Diagram

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

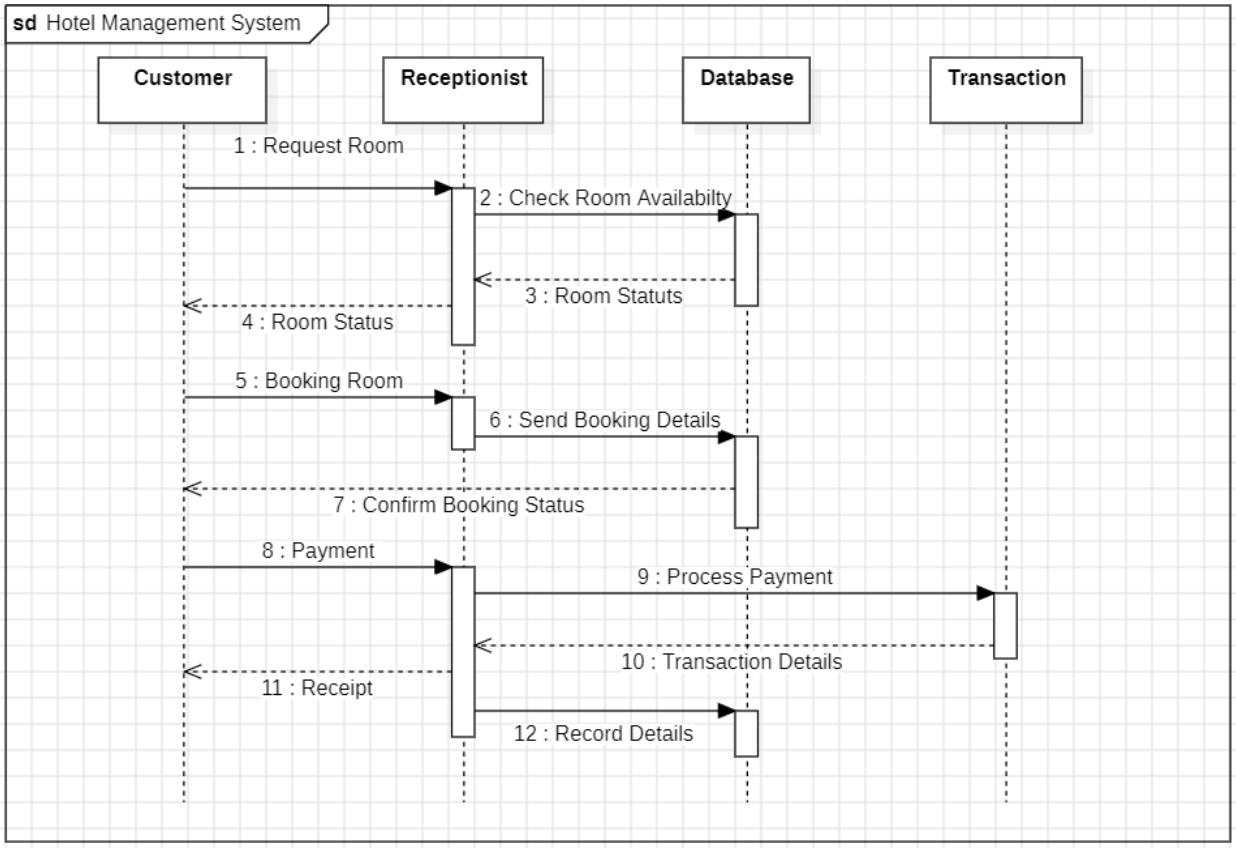


Fig1.4 Hotel Management System - Sequence Diagram

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

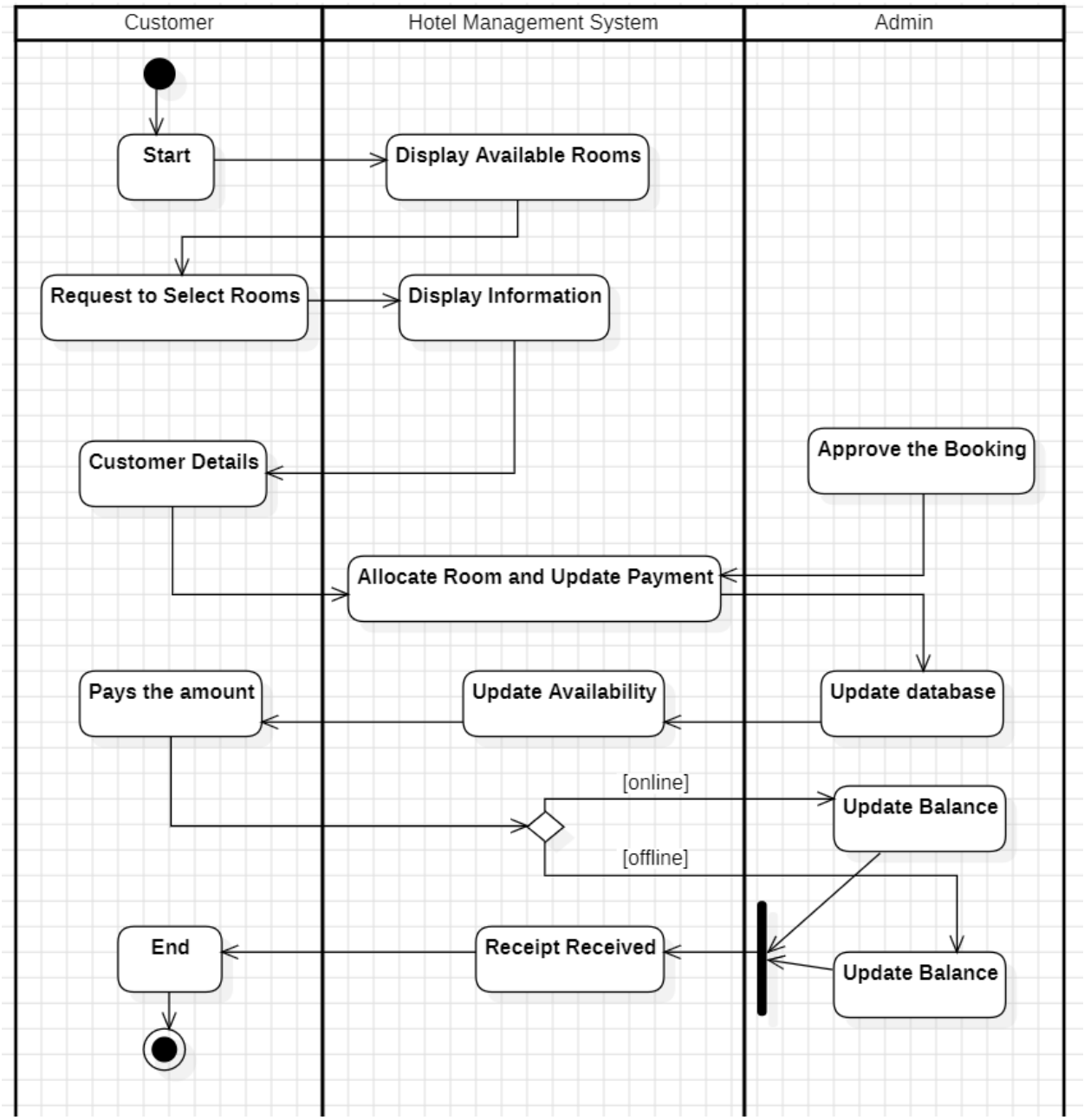


Fig1.5 Hotel Management System - Activity Diagram

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

LAB-1 30-9-24

SRS for credit card system

1. Introduction

1.1 purpose of this document

This software requirement specification SRS document outlines the requirements for the credit card system. It serves as a reference for stake holders, developers, and testers to ensure a clear understanding of the system's capabilities

1.2 Scope of this document

The credit card system is designed to manage the issuance, processing and payment of credit card transactions. This document covers functional & non-functional requirements design constraints, performance metrics.

1.3 overview

The credit card system will provide users with a secure and user-friendly interface for managing their credit accounts

2. General description

In this, general function includes credit card accounts, make Payments, and monitor transactions, consumers, financial institutions staff and system administrators

3. Functional requirements

This includes user Registration and authentication, credit card details, transaction management payment processing and account management

4. Interface requirements:

Application programming Interfaces (APIs): Interfaces for communication with external banking systems, fraud detection systems, and third party payment processors.

Data streams: Secure transmission of user data and transaction details through encrypted channels.

5. perform Requirements:

Response Time: The system should respond to user requests within 2 seconds

Transaction processing time: payments should be processed within 5 second

Memory usage: The system should efficiently utilize memory, aiming for less than 100 MB of RAM usage under Peak Load

6. Design constraints:

Technology stack: The system must be developed using java and Mysql

Security standard: compliance with pci dss is mandatory

7. Non-functional Attributes:

Security: Strong encryption for user data and transaction

Portability: The system should be accessible on various platforms including web and mobile

Reliability: the system should have 99.9% uptime and be fault tolerant.

8. preliminary schedule and budget

Timeline:

Requirements Gathering: 1 month

Development Phase : 3 months

Testing Phase : 1 month

Deployement : 1 month

Budget :

Deployement costs: $150,000

Testing and QA : $30,000

Deployment and maintenance. $20,000

30/9

Class Diagram

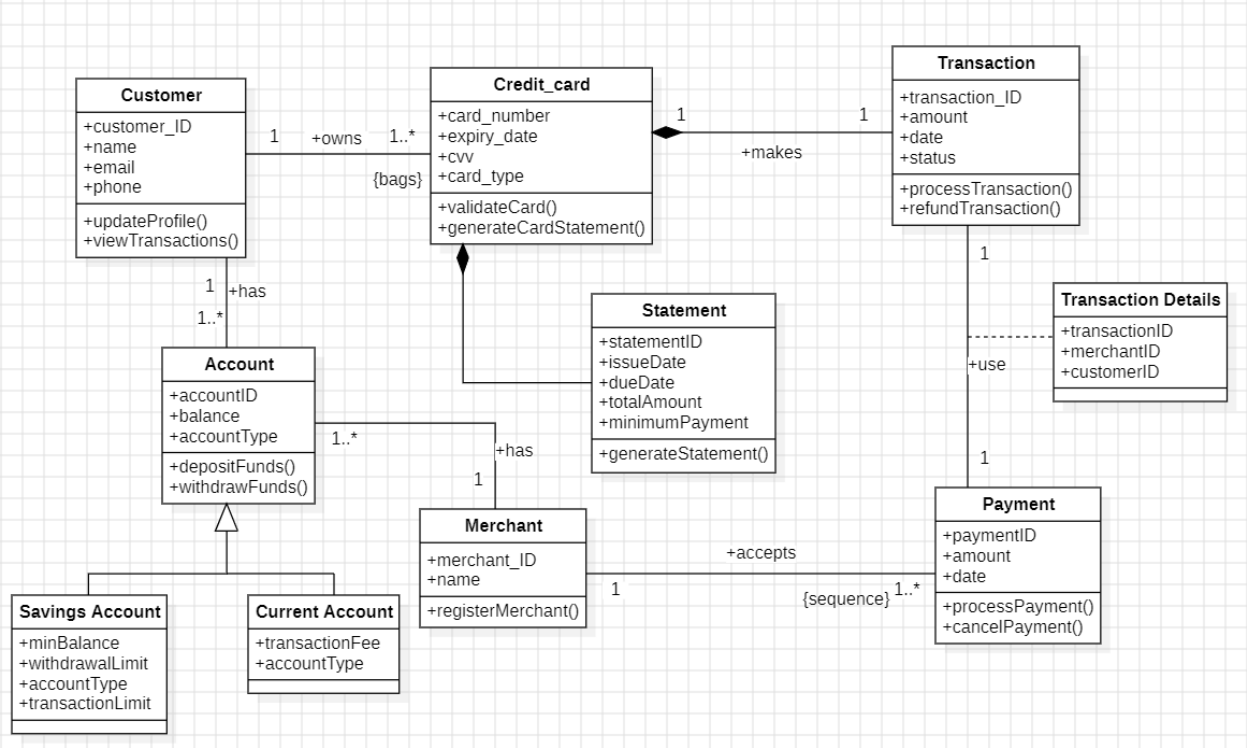


Fig 2.1 Credit Card Processing System - Class Diagram

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

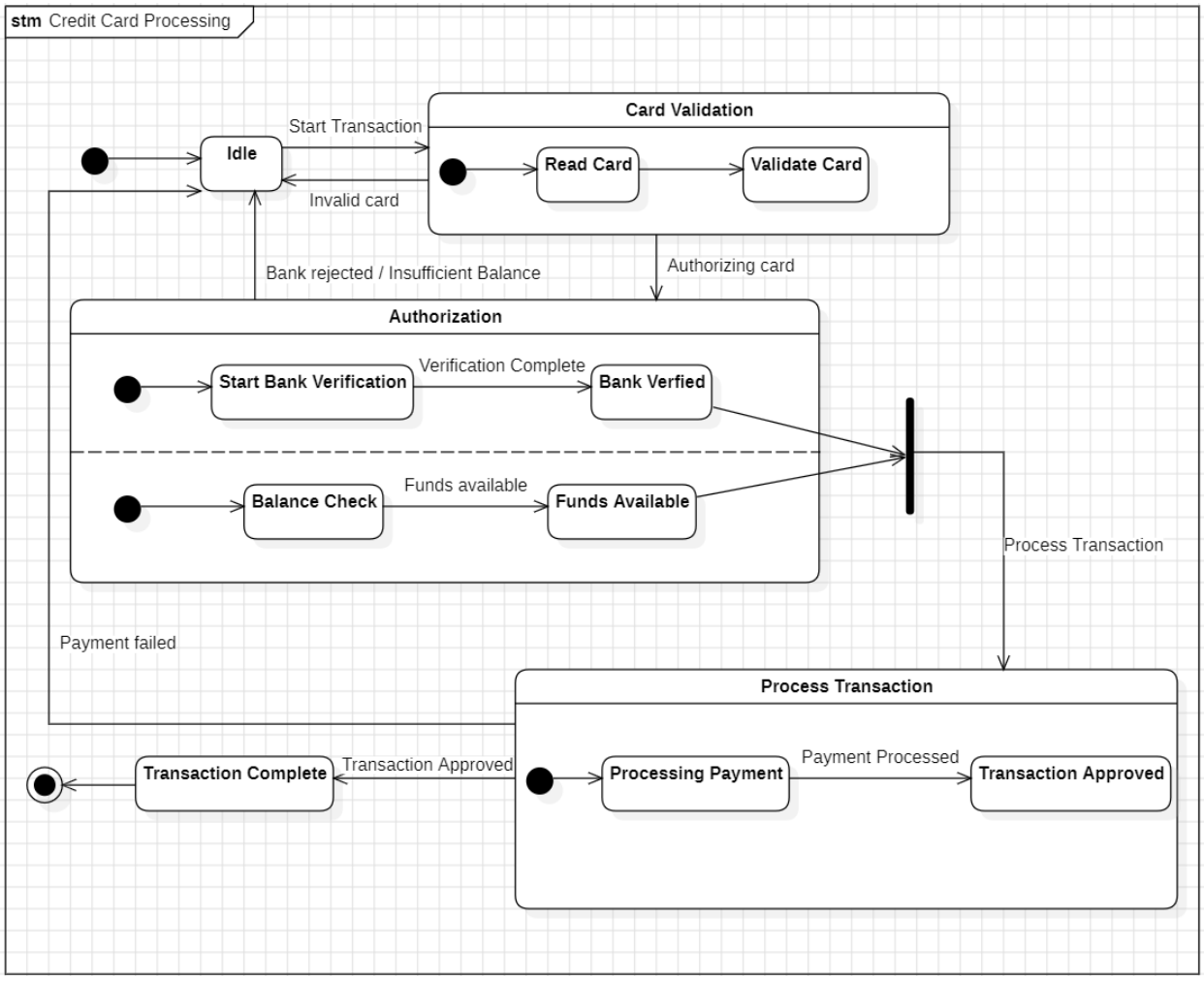


Fig 2.1 Credit Card Processing System - State Diagram

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

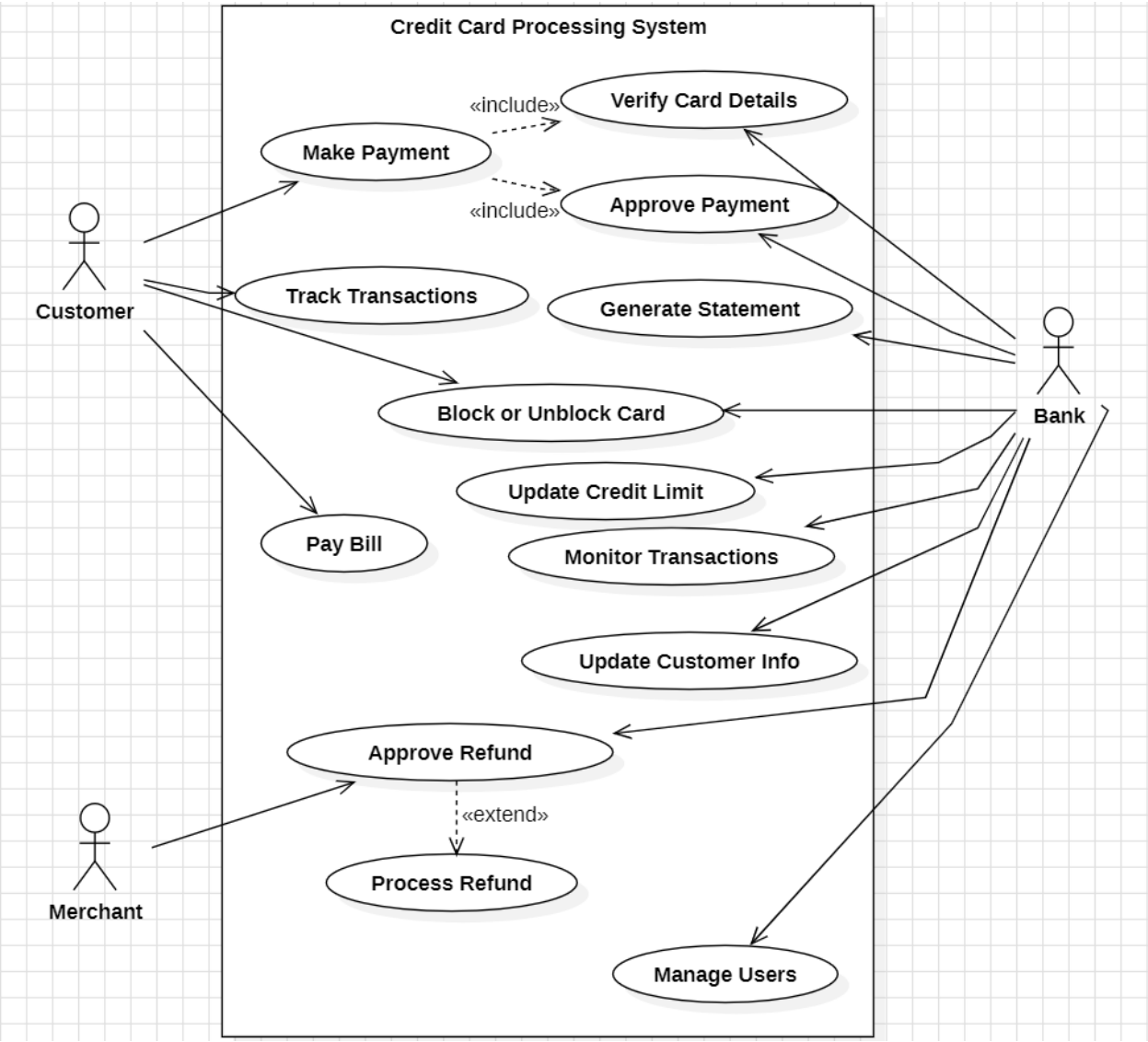


Fig 2.1 Credit Card Processing System - Use Case Diagram

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

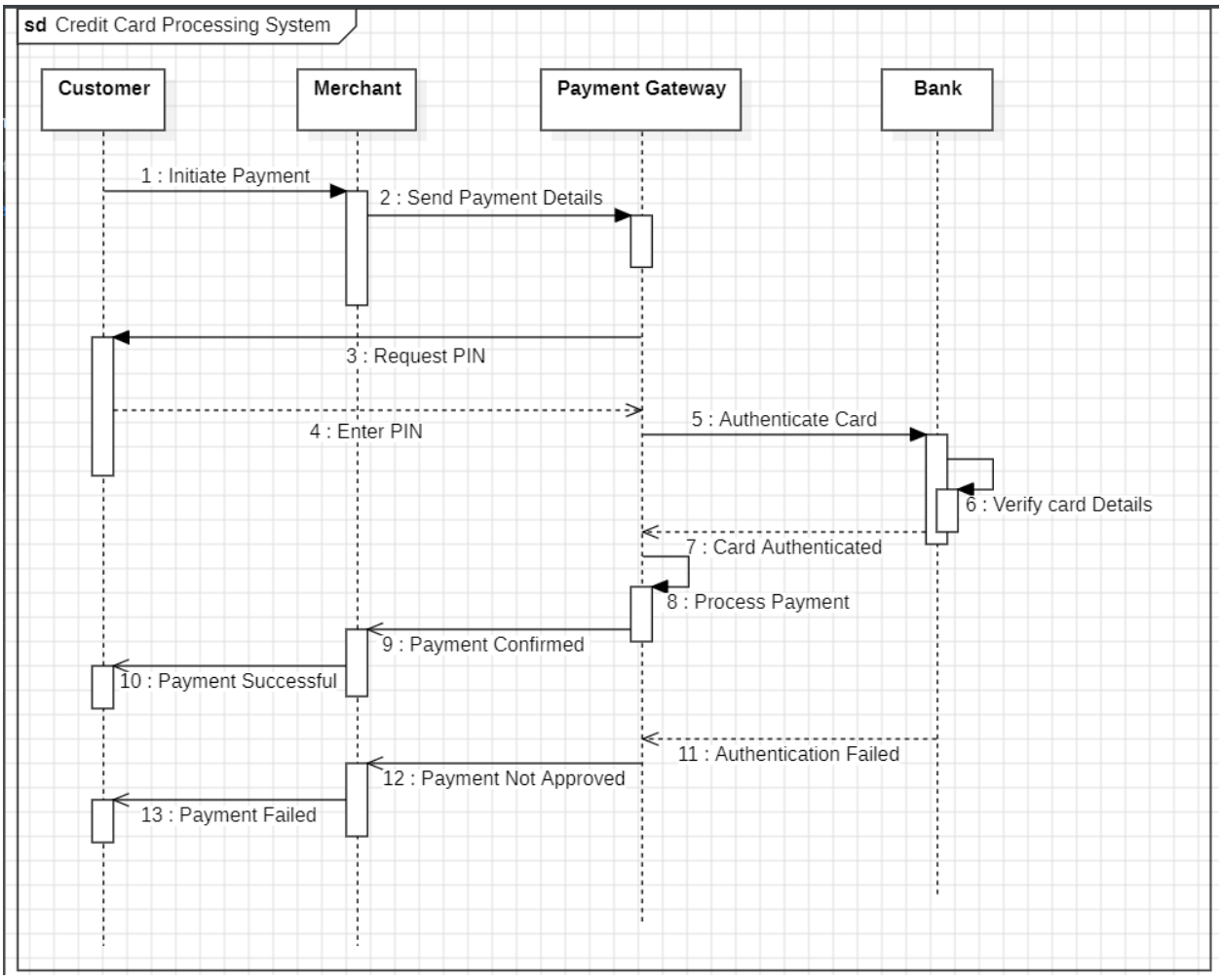


Fig 2.1 Credit Card Processing System - Sequence Diagram

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

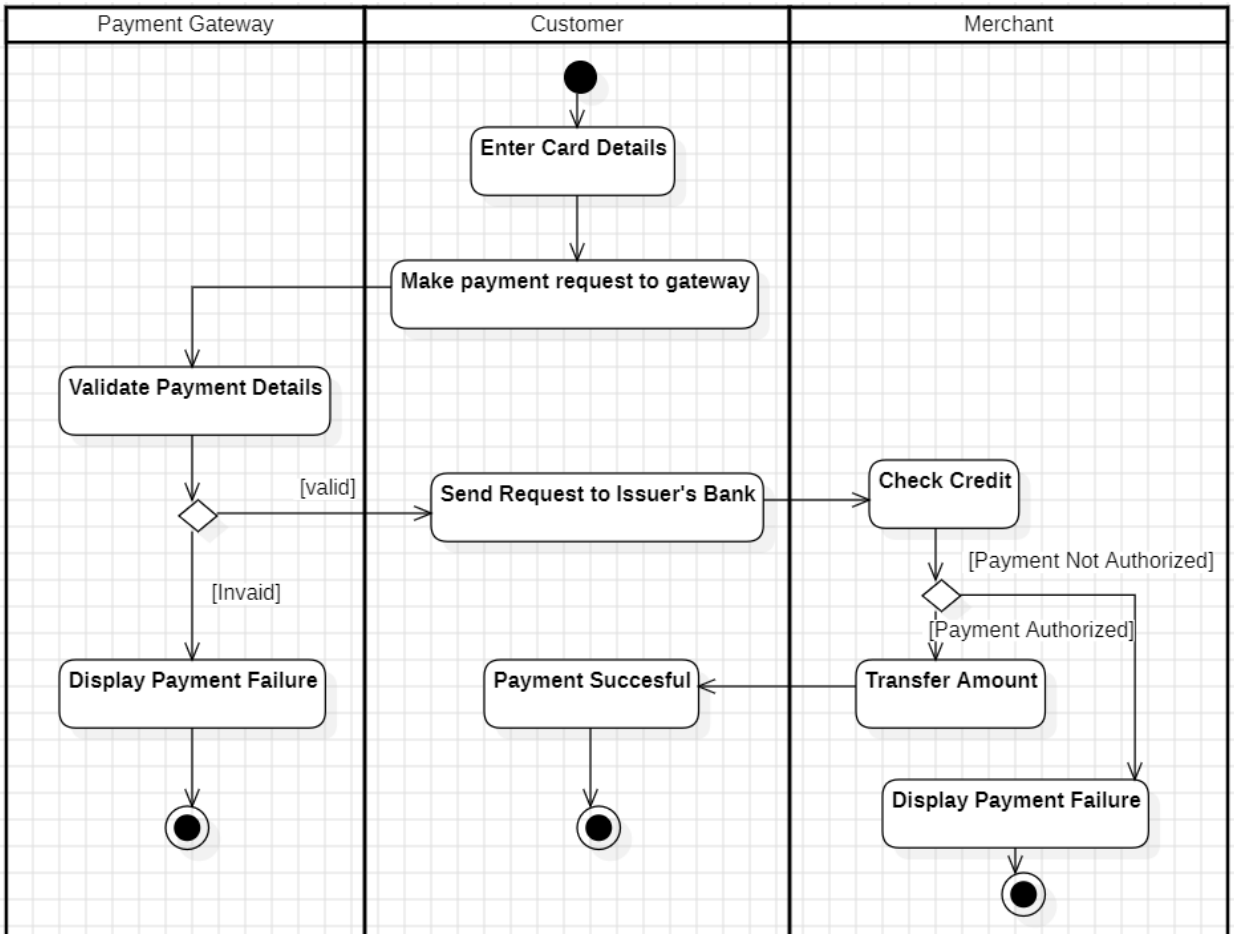


Fig 2.5 Credit Card Processing System - Activity Diagram

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

SRS for library management system

1 Introduction

1.1 Purpose of this document

This document outlines the requirements and specifications for the library management system (LMS) It aims to provide a comprehensive understanding of the system's functionalities

1.2 Scope of this document

The LMS is designed to automate the management of library resources including cataloging, member management and transaction processing

1.3 overview

The LMS will serve as a centralized platform for managing library operations, enhancing efficiency, improving user experience

2 General description

The LMS targets library staff and members, providing functionalities such as book cataloging, membership management, lending and returning books

3 Functional Requirements

3.1 User Registration & Authentication

3.2 Book Cataloging

3.3 Member Management

3.4 Book Lending

4. Interface Requirement

(i) User Interface (UI)

(ii) APIs

(iii) Database Integration

5. performance Requirements

(i) Response Time

(ii) Concurrent users.

(iii) Data storage

6. Design Constraints

(i) platform

(ii) Database

(iii) Framework

7. Non-functional Attributes

(i) Security

(ii) Reliability

(iii) Scalability

8. Preliminary schedule and Budget

(i) Estimated Development Duration

(ii) Estimated Budget

Class Diagram

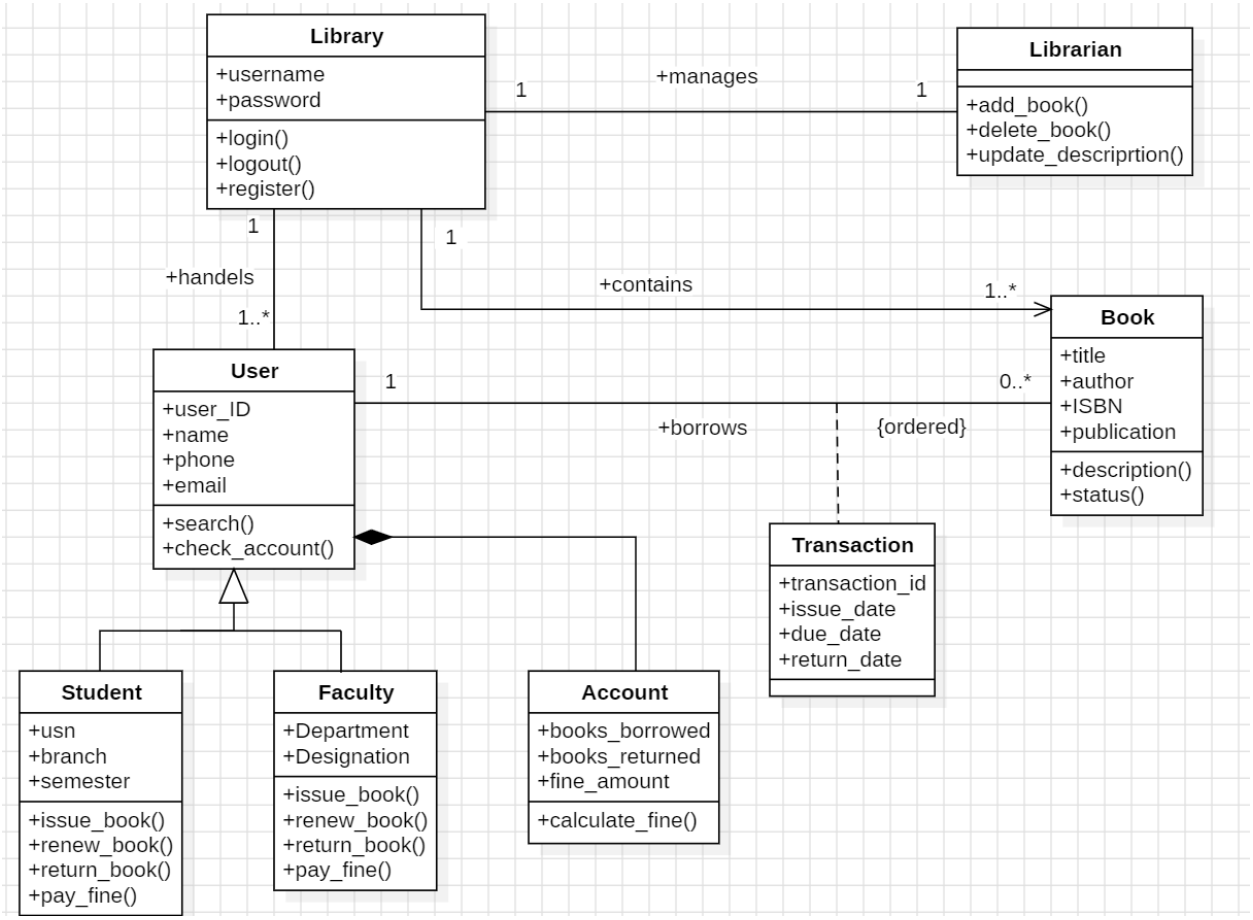


Fig 3.1 Library Management System -Class Diagram

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

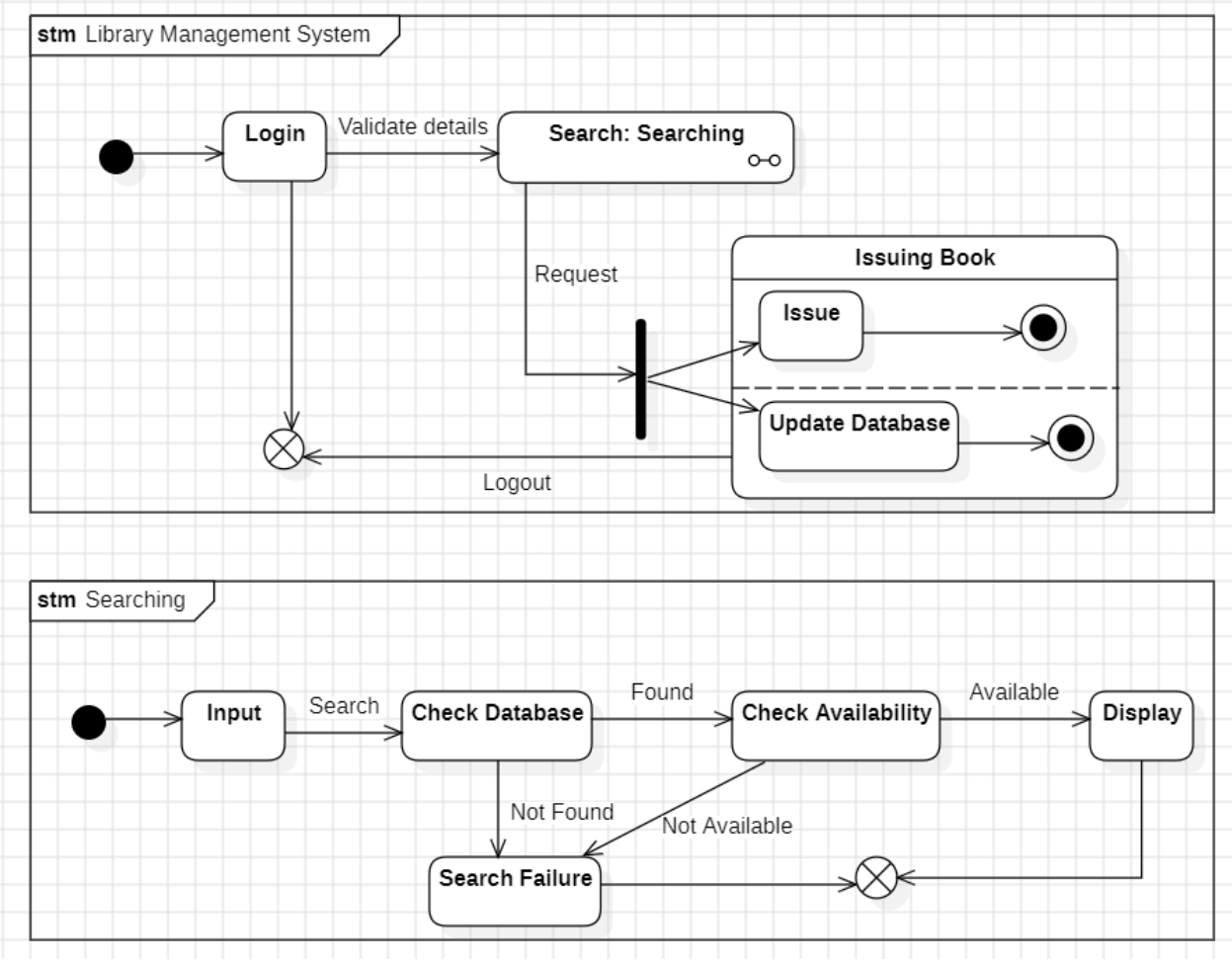


Fig 3.2 Library Management System - State Diagram

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

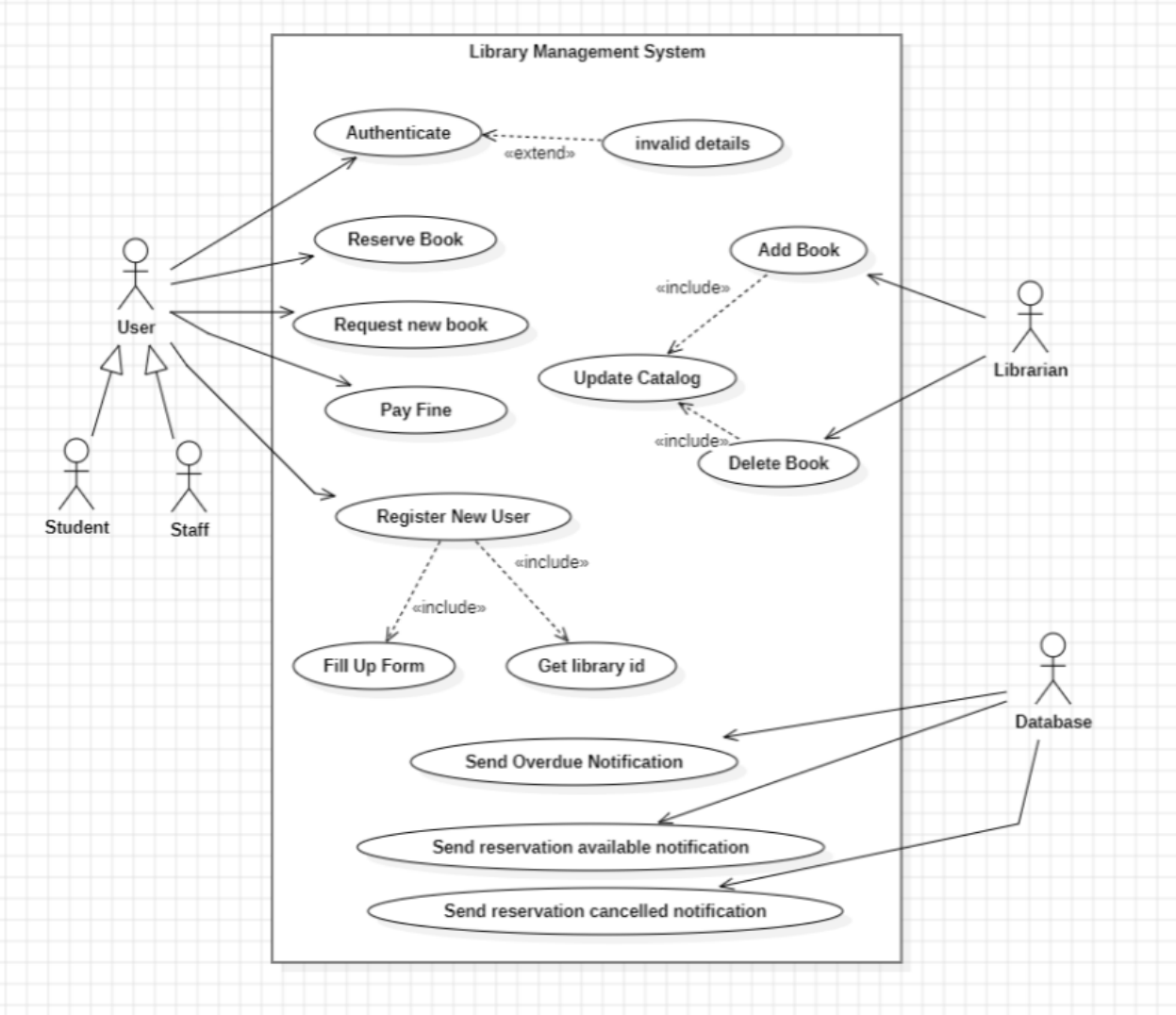


Fig 3.3 Library Management System - Use Case Diagram

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

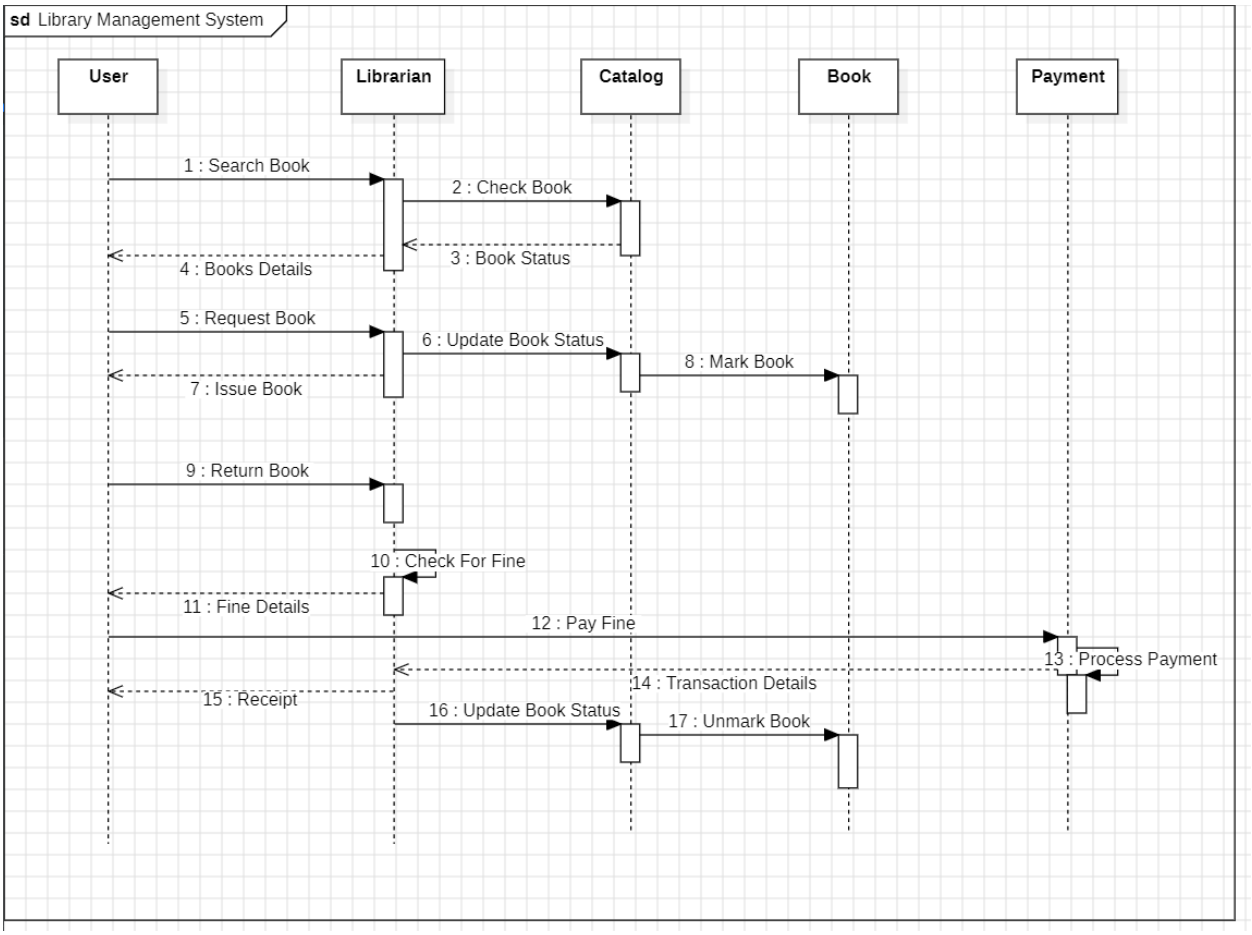


Fig 3.4 Library Management System - Sequence Diagram

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

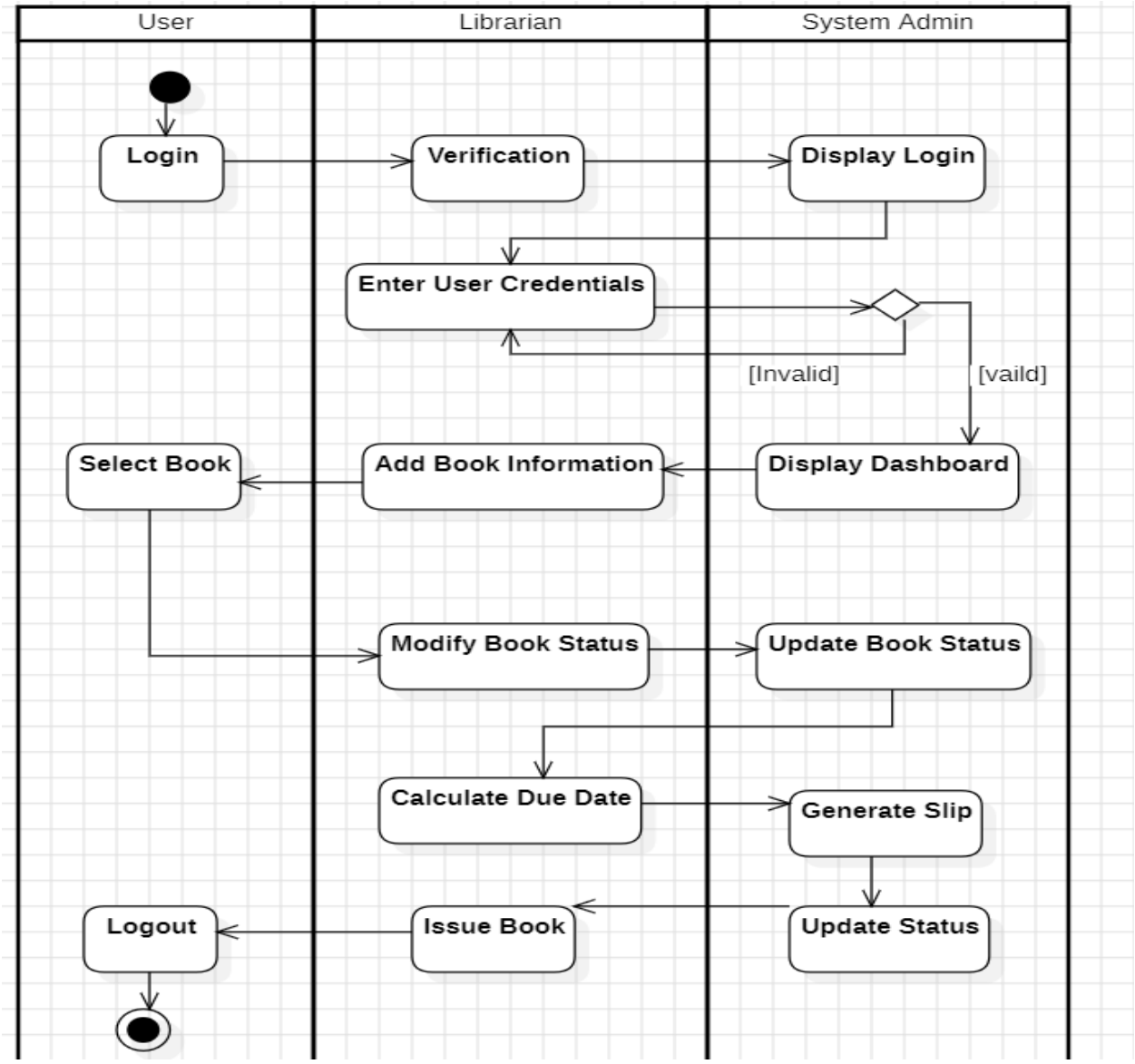


Fig 3.5 Library Management System - Activity Diagram

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. Finally, the system administrator updates the overall book status, and the user can log out. The diagram illustrates the interconnected roles and actions of users, librarians, and system administrators in the library's book borrowing process.

4. Stock Maintenance System

Software Requirement Specification

→SRS for Stock maintenance system

1. Introduction

1.1 purpose of this Document:

The purpose of this document is to outline the functional and nonfunctional requirements of the stock maintenance system. This document serves as a guide for the development team to understand the system's requirements.

1.2 Scope of this Document

This document covers the scope of the SMS, detailing how it will be handled stock tracking, updates, sales, purchases, and inventory management for a business

1.3 Overview

The stock maintenance system is designed to automate the process of managing inventories in retail and wholesale business. It will track the stock levels of products, record sales and purchases, notify users when stock levels are low and generate reports for better decision-making

2. General description:

The SMS will allow users to manage stock levels, track incoming and outgoing products and generate detailed reports on stock usage and availability. It will target retail business, where sales

3. Functional requirements

(i) Stocking tracking: The system should allow users to add, update and delete stock records

(ii) Sales and purchases: Users should be able to record sales and Purchases transaction

4. Interface requirements

(i) User interface : The system will feature a web-based user interface accessible from desktops

(ii) Software Interface : The system will interact with a relational database

5. Performance Requirements

(i) Speed : The system should handle up to 500 concurrent users without noticeable performance degradation

(ii) Memory : The system should efficiently use the memory

6 Design constraints

(i) Algorithm constraints : The system must implement efficient algorithms for searching, filtering and sorting stock.

(ii) Hardware constraints : The system should run on standard web servers with at least 8 GB of RAM and 4 CPU cores

7. Non-functional requirements

(i) Scalability : The system should be scalable to handle large inventories and user bases as the business grows

(ii) Data integrity : Data should be automatically backed up daily to prevent data loss

8. Preliminary schedule and Budget

Schedule :

(i) Requirements Gathering : 2 weeks

(ii) System Design: 1 month

(iii) Development and testing: 3 months

Budget

(i) Development: 20,000₦

(ii) Testing: 5000₦

(iii) Deployement: 5000₦

Class Diagram

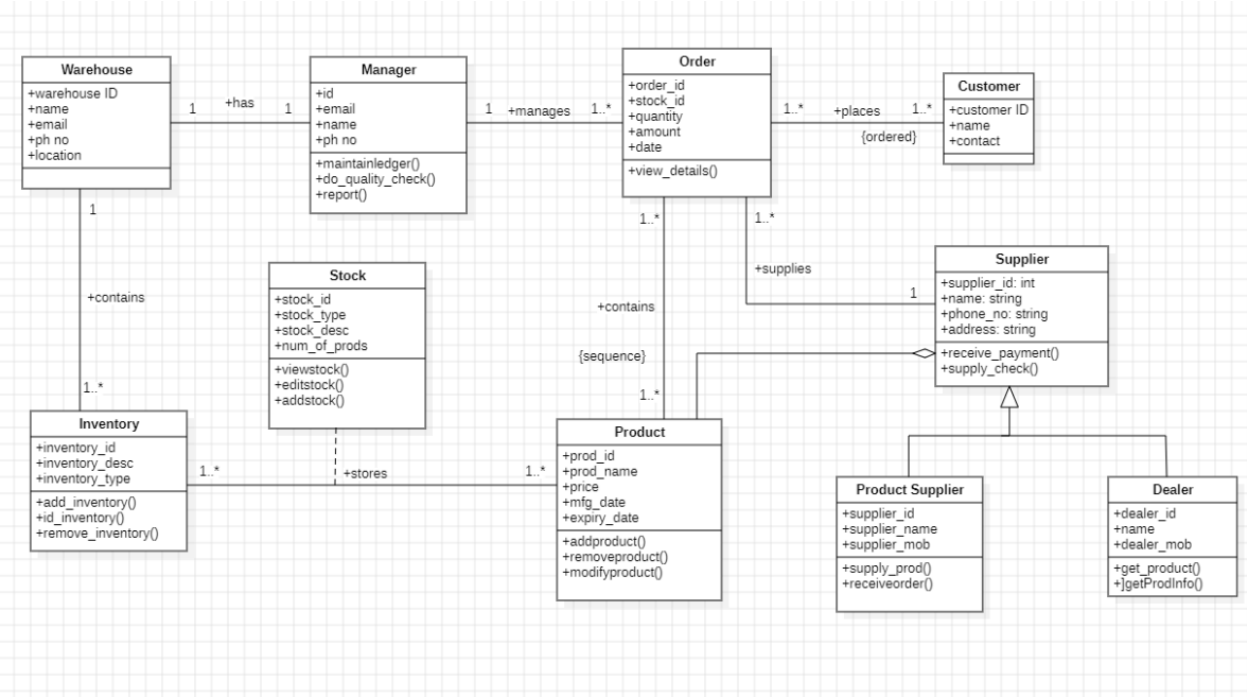


Fig 4.1 Stock Maintenance System - Class Diagram

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

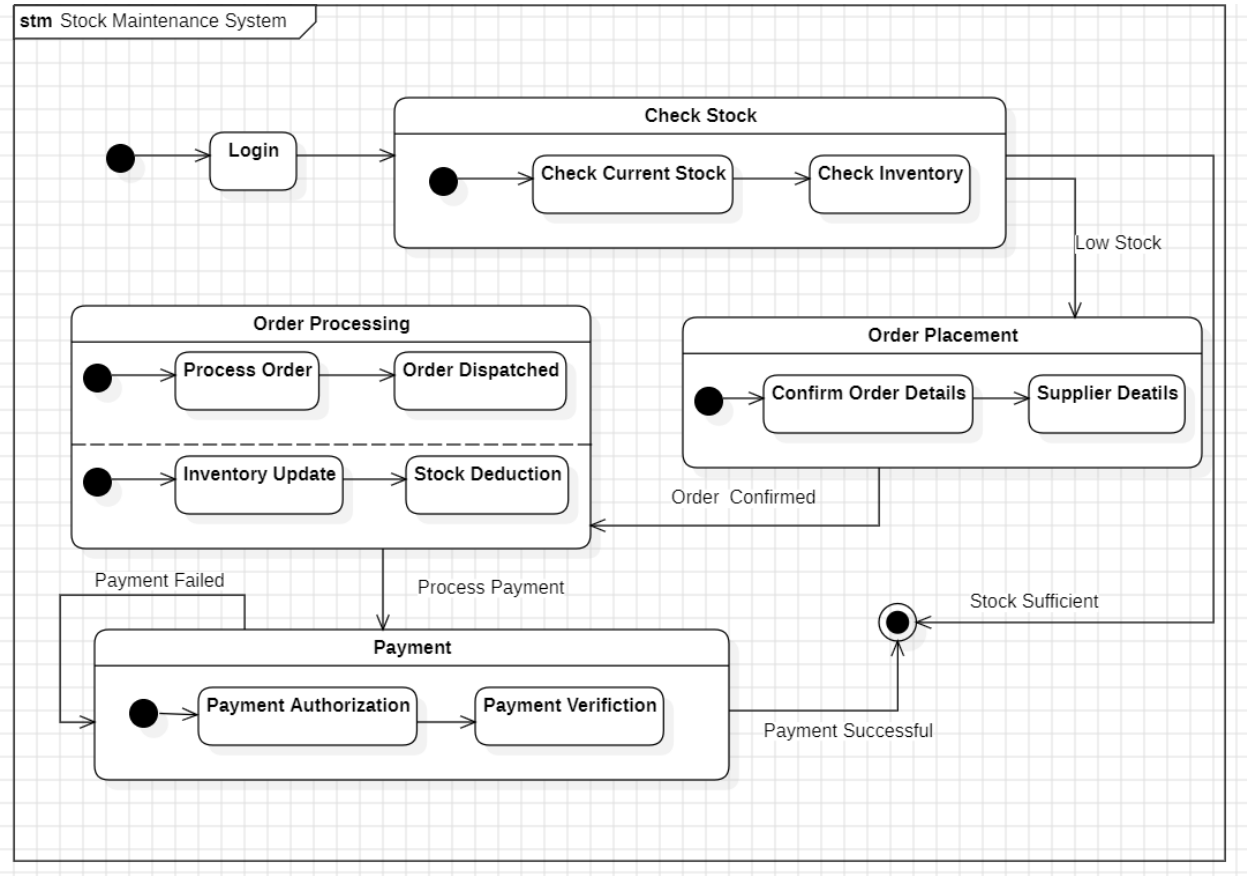


Fig 4.2 Stock Maintenance System - State Diagram

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

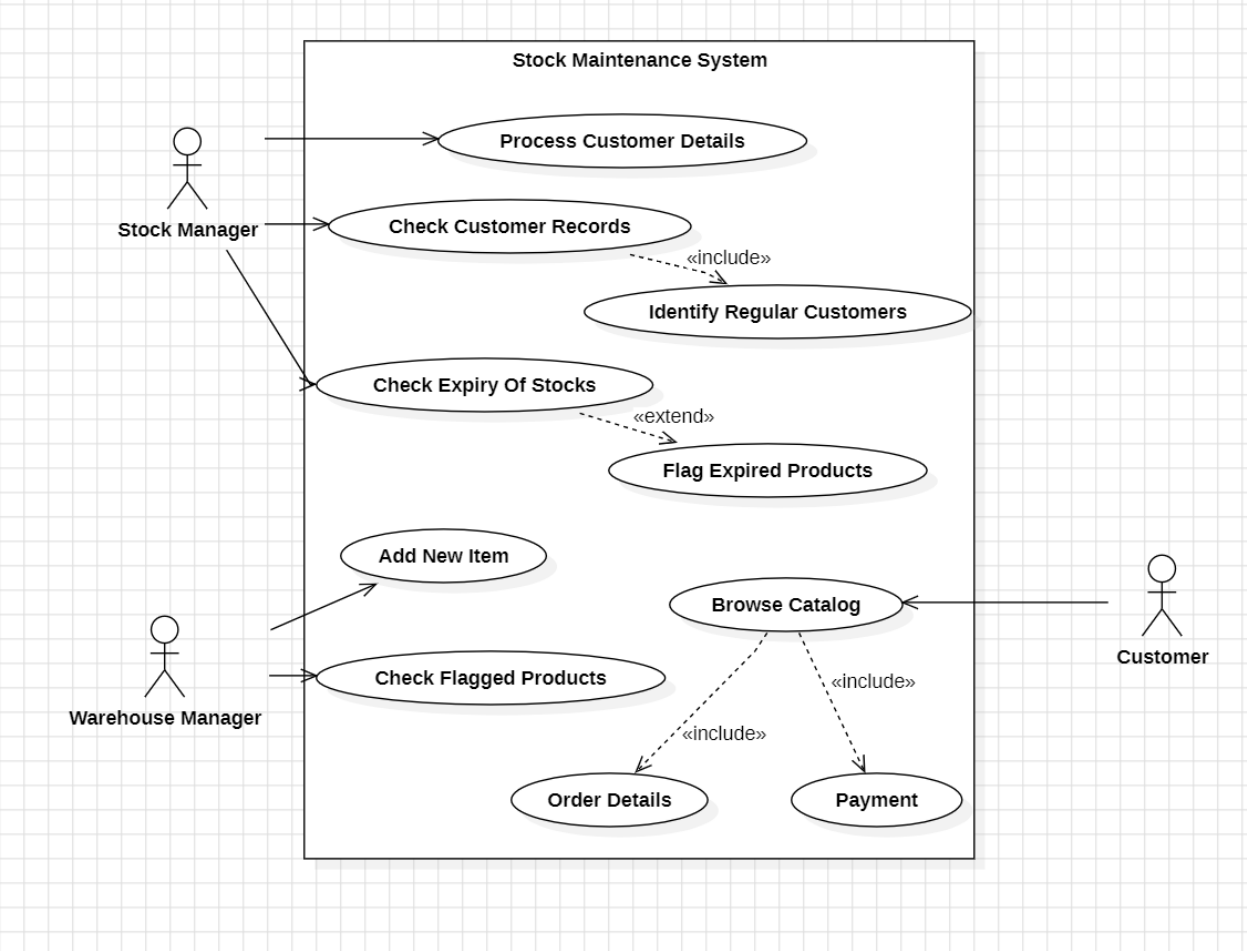


Fig 4.3 Stock Maintenance System - Use Case Diagram

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

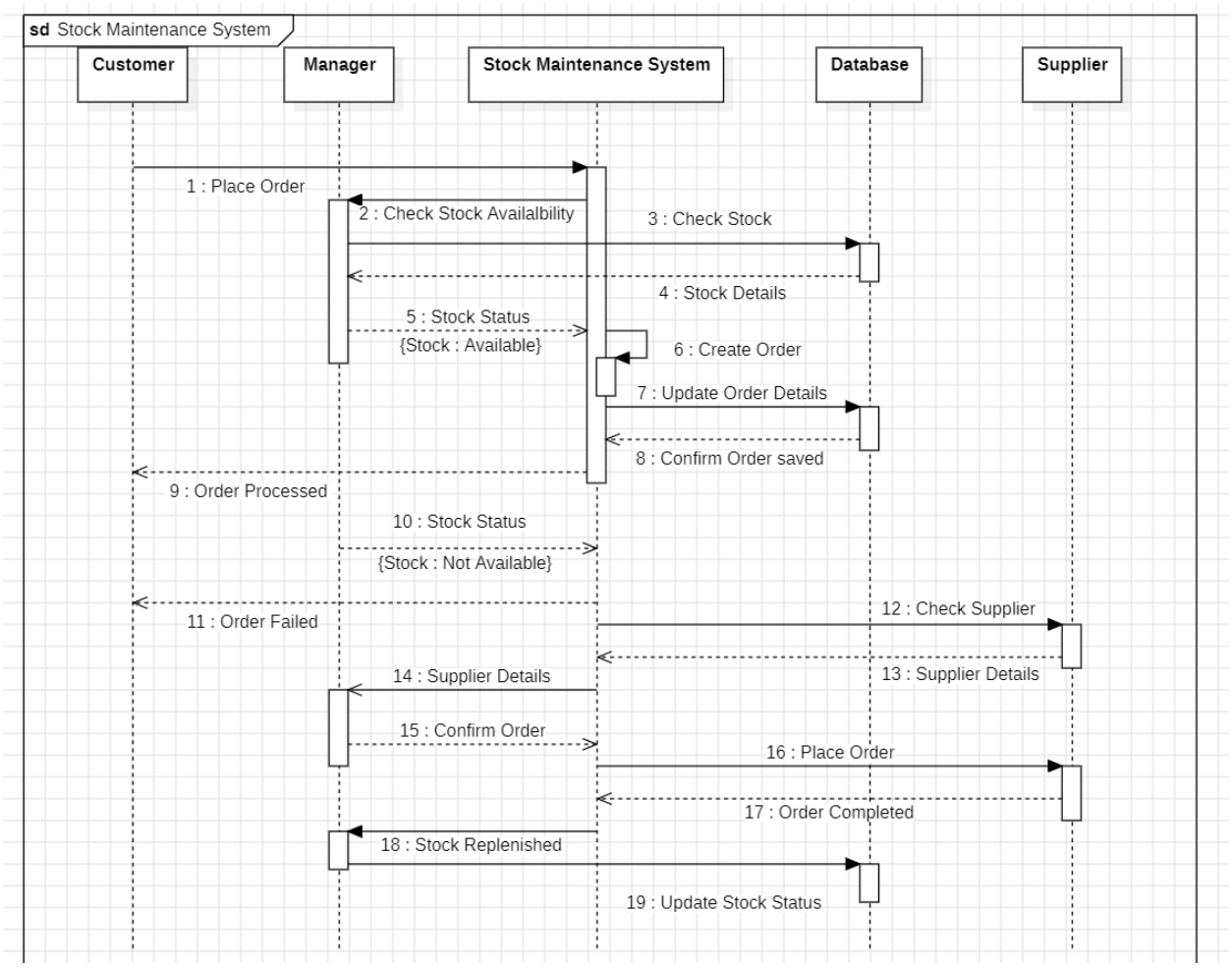


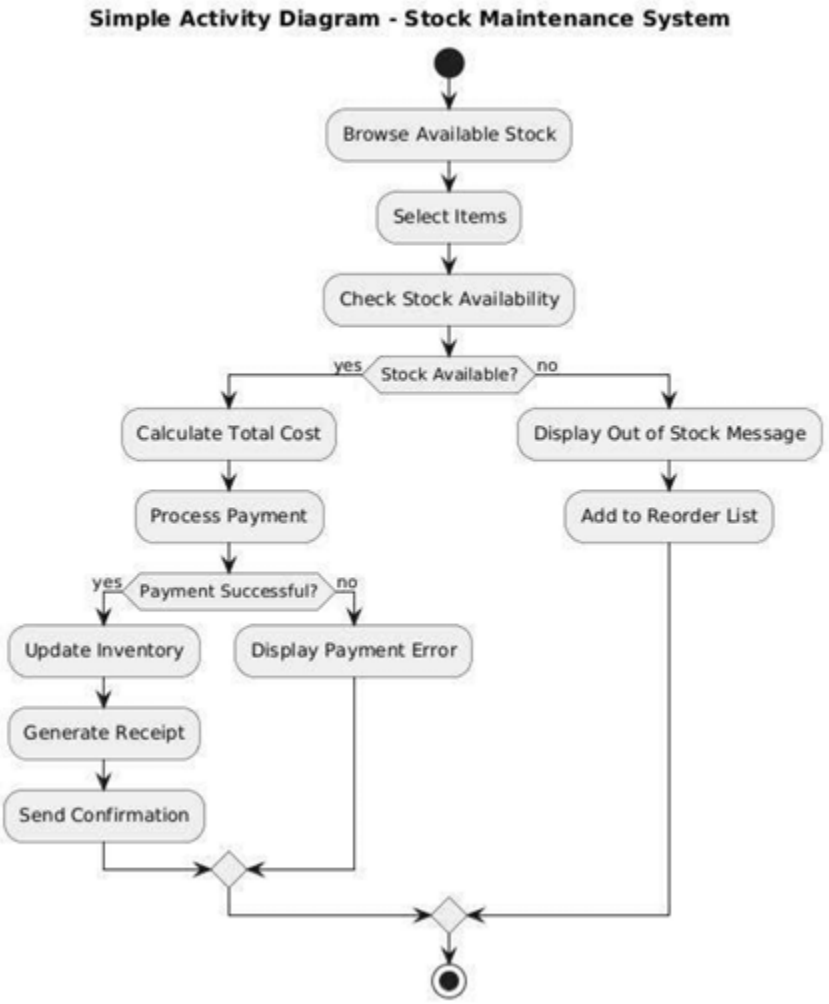
Fig 4.4 Stock Maintenance System - Sequence Diagram

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

Fig 4.5 Stock Maintenance System - Activity Diagram



5. Passport Automation System

Software Requirement Specification

→SRS for passport Automation System

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to define the requirements for the passport Automation system (PAS). The system aims to streamline the process of passport application and renewal

1.2 Scope of the Document

This document encompasses the development of an automated system for managing passport-related activities, including application submission, document verification

1.3 Overview

The passport Automation system will allow users to submit passport applications, upload necessary documents, and track the status of their application

2 General Description

The PAS will automate various functions such as application processing document verification, interview scheduling and passport issuance

3 Functional Requirements

(i) Document verification: officials should have access to a dashboard to review and verify the submitted documents

(ii) passport issuance: upon approval, the system should generate a passport ID and print-ready passport format

4. Interface Requirements

(i) User Interface: The system will feature a web based interface for both applicants and officials

(ii) Hardware interface: The system must be deployed on secure government servers with proper authentication

5. Performance Requirements

(i) Speed: The system should be capable of processing up to 1000s applications.

(ii) memory: The system should efficiently handle large document uploads

6. Design constraints

(i) Algorithm constraints: The should use secure algorithms for encryption and decryption of sensitive data such as personal identification

(ii) Software constraints: The system should be built using government approved software technologies

7. Non functional requirements

(i) security: The system must be strong authentication mechanism such as multi-factor authentication

8. preliminary Schedule and Budget

Schedule

(i) Requirement Gathering: 1 month

(ii) System design: 1 month

(iii) Deployment: 1 month

Budget

(i) Development: 70,000 N

(ii) Deployment: 10000 N

Class Diagram

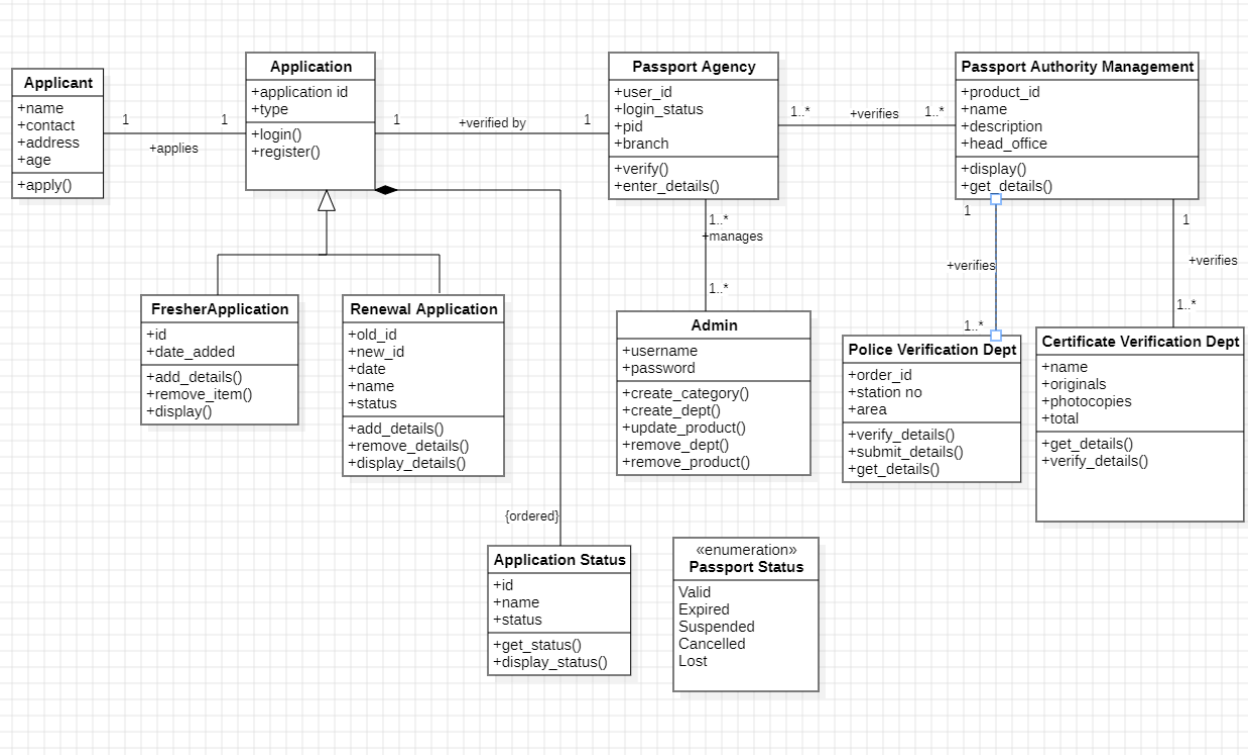


Fig 5.1 Passport Automation System - Class Diagram

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

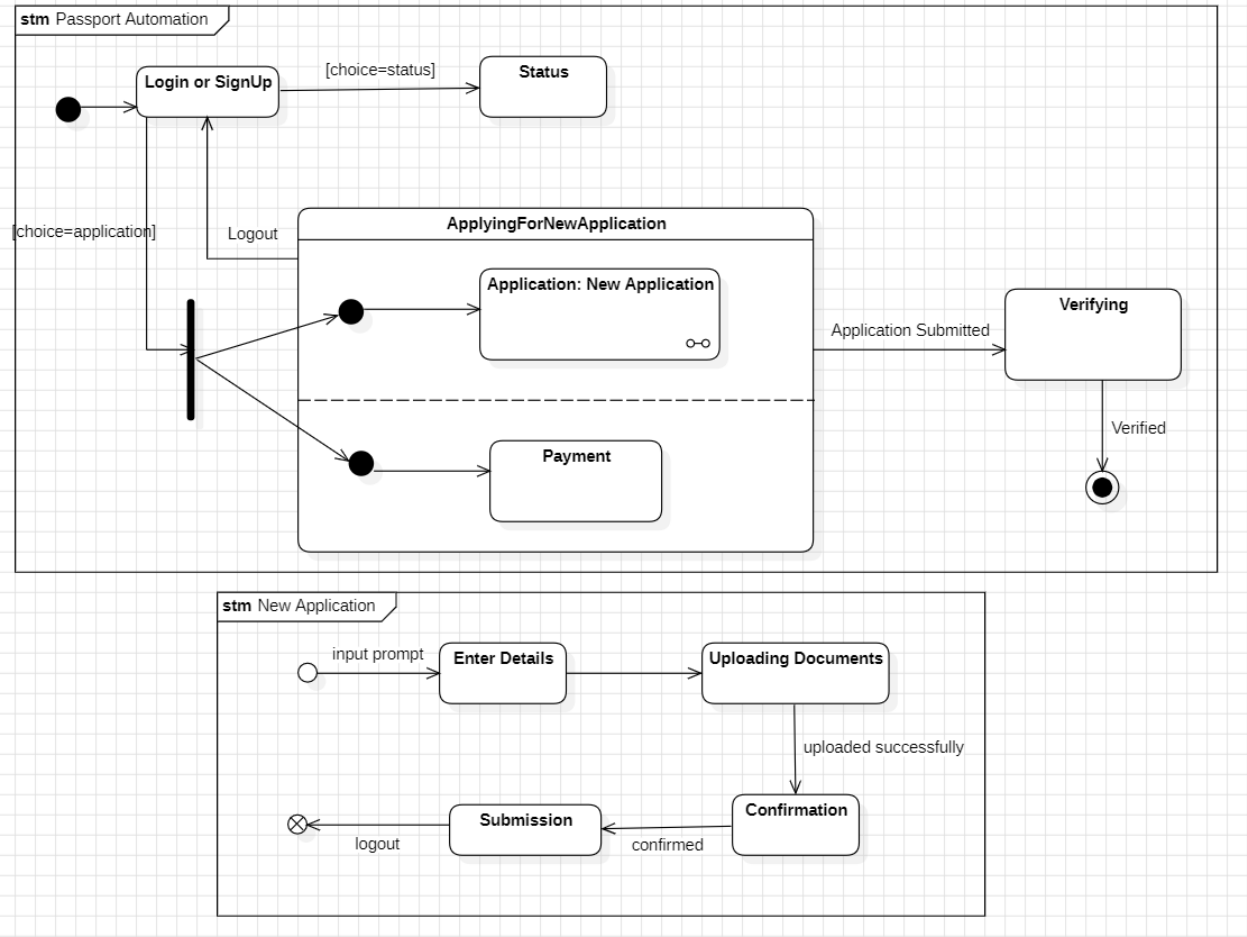


Fig 5.2 Passport Automation System - State Diagram

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

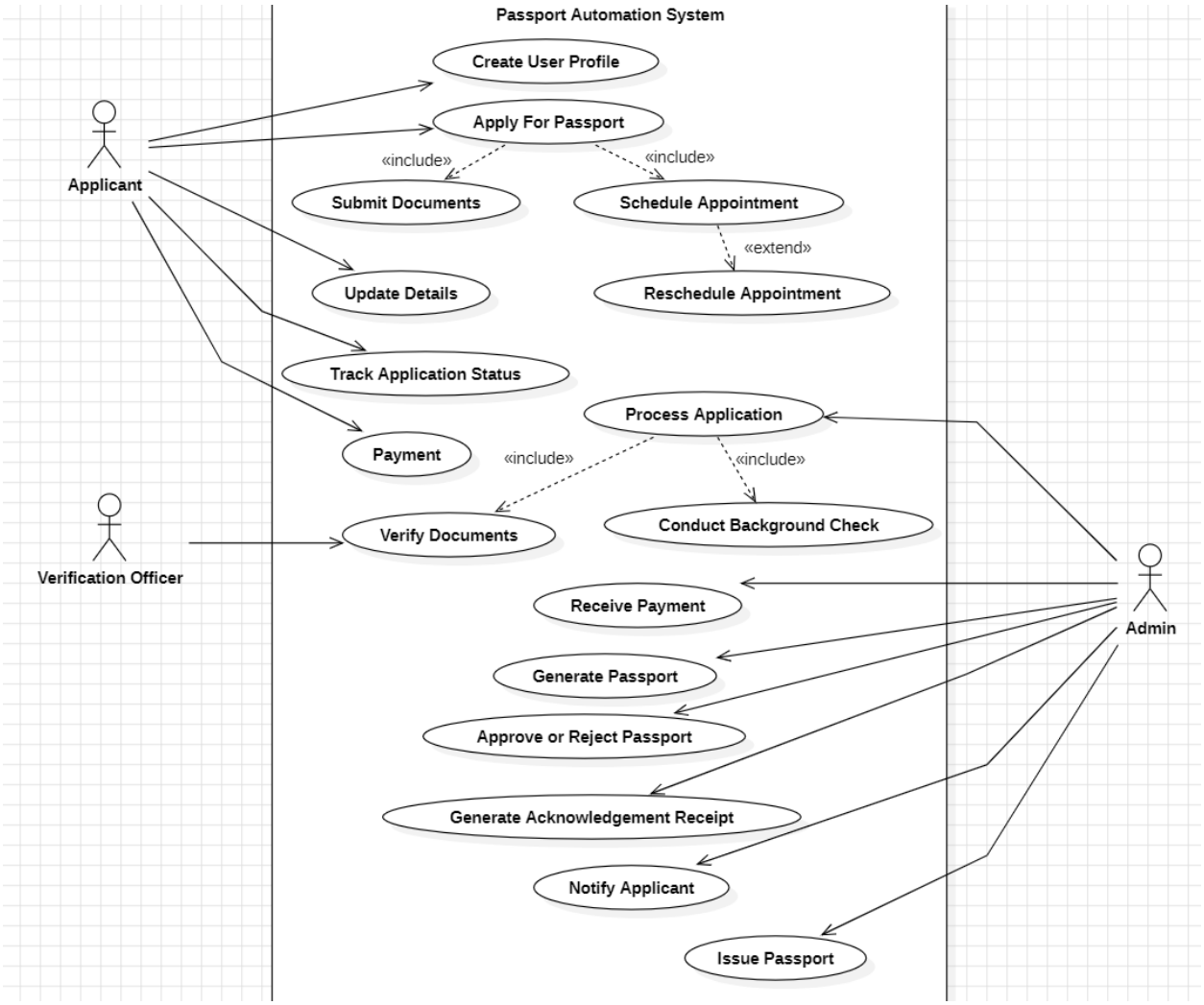


Fig 5.3 Passport Automation System - Use Case Diagram

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

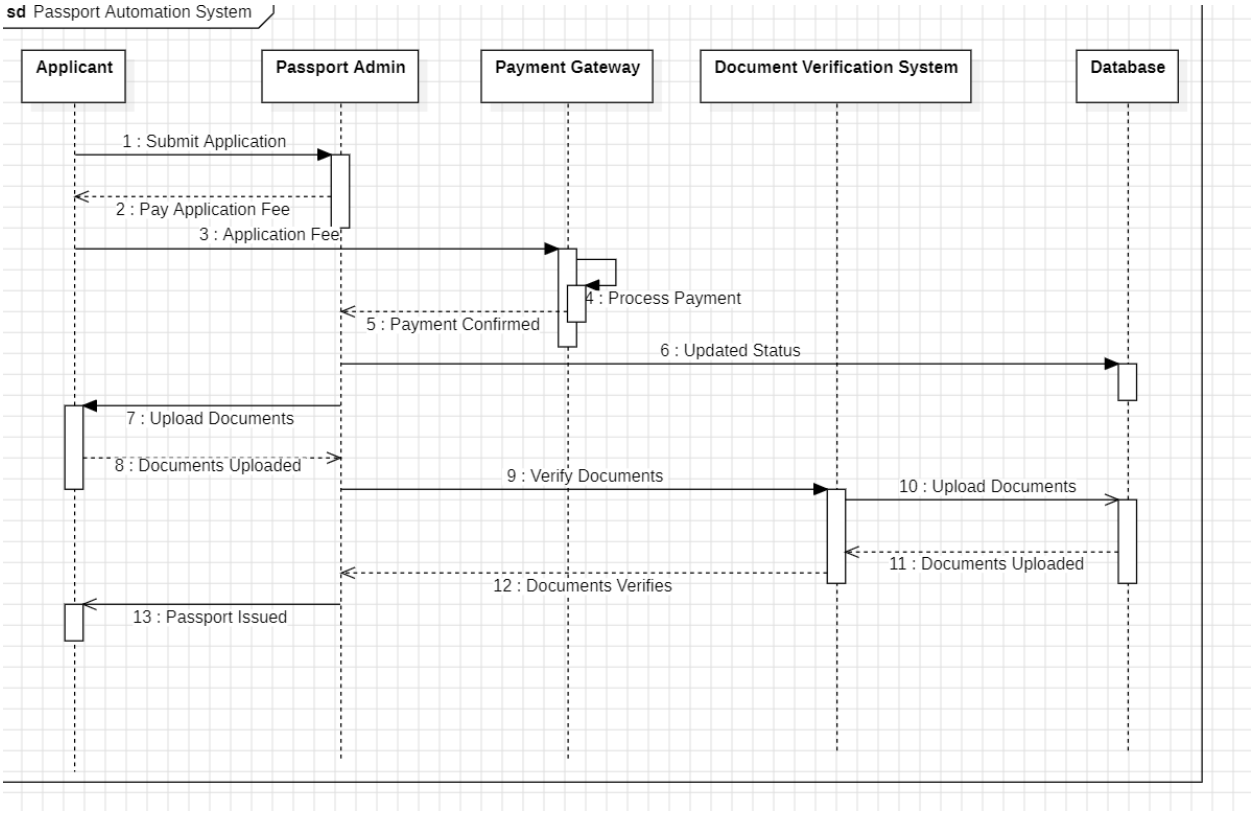


Fig 5.4 Passport Automation System - Sequence Diagram

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

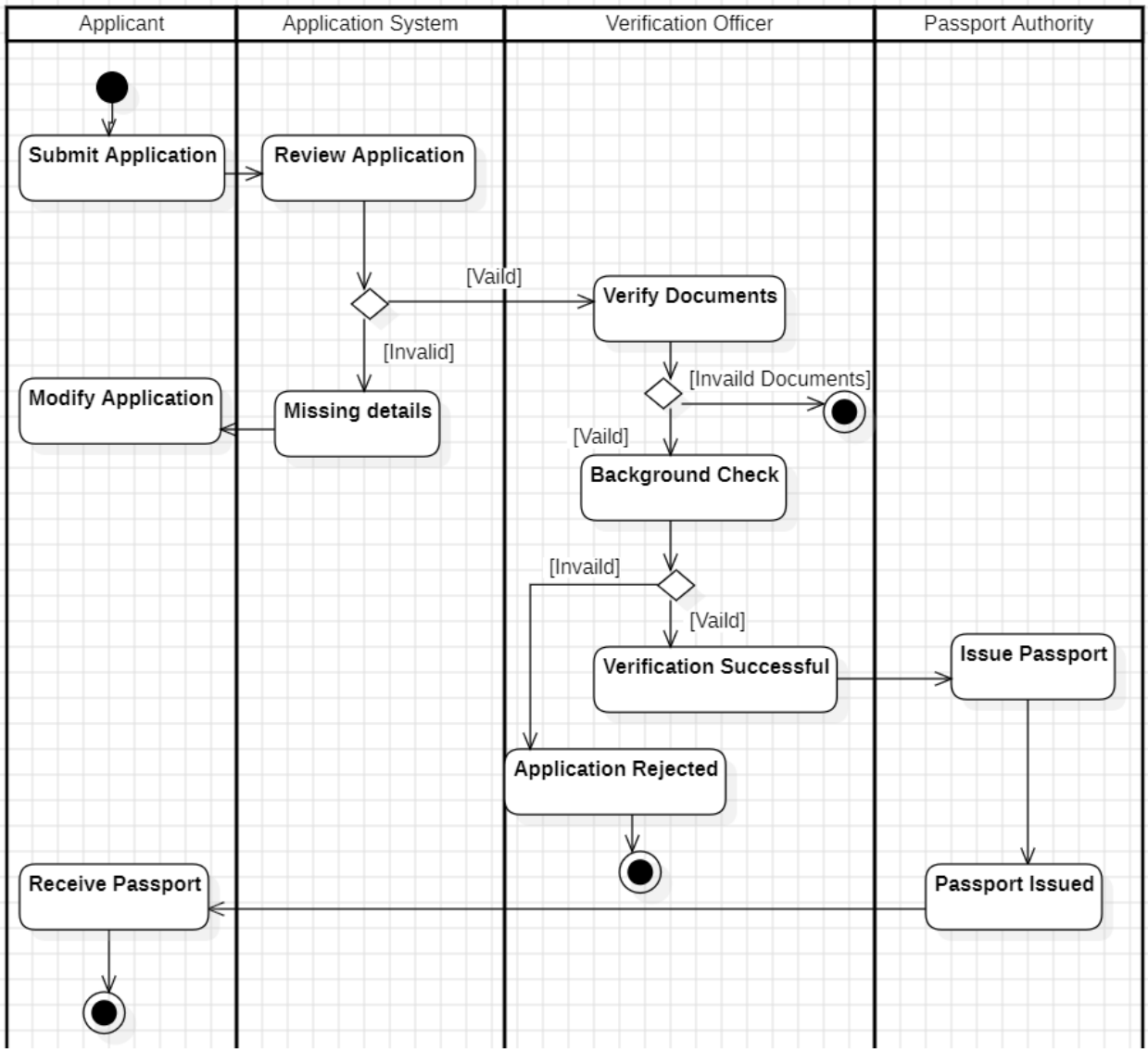


Fig 5.5 Passport Automation System - Activity Diagram

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.