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:

Aditya Kumar

1BM22CS018

Department of Computer Science and Engineering

B.M.S. College of Engineering

Bull Temple Road, Basavanagudi, Bangalore 560 019

Sep-Jan 2025

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 Aditya Kumar (1BM22CS018) during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

NAME OF THE FACULTY: Sandhya A Kulkarni

Department of Computer Science and Engineering

B.M.S. College of Engineering, Bangalore

Table of Contents

1. Hotel Management System

Problem statement

The hotel management system aims to automate and streamline the operations of a hotel. It handles room booking, check-ins, check-outs, and billing processes, ensuring efficient service delivery to customers. The system should provide features such as room availability status, staff management, and customer records while minimizing manual errors and improving operational efficiency.

SRS Document

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to outline the requirements for developing a Hotel Management System. It will provide a clear understanding of the system's functionalities, constraints, and objectives to ensure that all stakeholders are aligned on the development goals. This document serves as a guide for developers, testers, and clients to ensure the successful delivery of the system.

1.2 Scope of the Document

The Hotel Management System will automate hotel operations, including room reservations, check-ins, check-outs, billing, and staff management. The system will cater to both guests and staff, enhancing operational efficiency and customer satisfaction. The project is estimated to take six months to develop, with an approximate budget of $50,000.

1.3 Overview

The Hotel Management System is a web-based application that streamlines hotel operations by integrating various functionalities into a single platform. The system provides user-friendly interfaces for both hotel staff and guests, allowing for seamless management of reservations, inventory, billing, and customer interactions. This reduces manual errors and improves service delivery.

2. General Description

The system aims to:

- Simplify hotel operations by automating tasks such as bookings, payments, and inventory management.

- Provide a user-friendly interface for hotel staff and guests.
- Enhance customer satisfaction through efficient service delivery.

Features and Benefits:

- Centralized system for room availability and bookings.
- Automated billing and payment processes.
- Easy access to guest history and preferences for personalized service.
- Comprehensive reporting for performance analysis.

User Characteristics:

- **Guests:** Access the system to book rooms, check-in/out, and make payments.
- **Staff:** Manage reservations, room assignments, inventory, and customer interactions.
- **Admins:** Oversee the system, manage staff access, and generate performance reports.

3. Functional Requirements

- **Room Booking and Reservation:** Guests can search for available rooms, view details, and book online.
- **Check-in and Check-out:** Staff can manage guest check-ins and check-outs efficiently.
- **Billing System:** Generate invoices and process payments through integrated payment gateways.
- **Staff Management:** Admins can assign roles and manage staff information.
- **Reporting:** Generate reports on occupancy rates, revenue, and customer feedback.
- **Notifications:** Send automated email or SMS alerts for booking confirmations and reminders.

4. Interface Requirements

- **User Interface:** A web-based platform accessible on desktops, tablets, and smartphones.
- **Database Interface:** Integration with SQL databases to store and retrieve guest, booking, and billing information.

- **Payment Gateway Interface:** Secure integration with third-party payment systems like PayPal or Stripe.

- **API Integration:** Connect with external systems for room inventory and pricing updates.

5. Performance Requirements

- The system must handle up to 500 simultaneous users without performance degradation.
- Room search results should be displayed within 2 seconds.
- The maximum allowable error rate for payment processing is 0.1%.
- The system must support 24/7 availability with 99.9% uptime.

6. Design Constraints

- Must be compatible with Windows, macOS, and major web browsers (Chrome, Firefox, Safari).
- Use of open-source frameworks and tools to minimize costs.
- Follow security standards such as PCI DSS for payment processing.
- Support multi-language functionality for global use.

7. Non-Functional Attributes

- **Security:** Secure user data with encryption and multi-factor authentication.
- **Portability:** Ensure the system can run on different devices without additional setup.
- **Reliability:** Achieve a mean time between failures (MTBF) of 10,000 hours.
- **Scalability:** Support future expansions like adding more hotels or features.
- **Data Integrity:** Prevent data loss during system updates or crashes

8. Preliminary Schedule and Budget

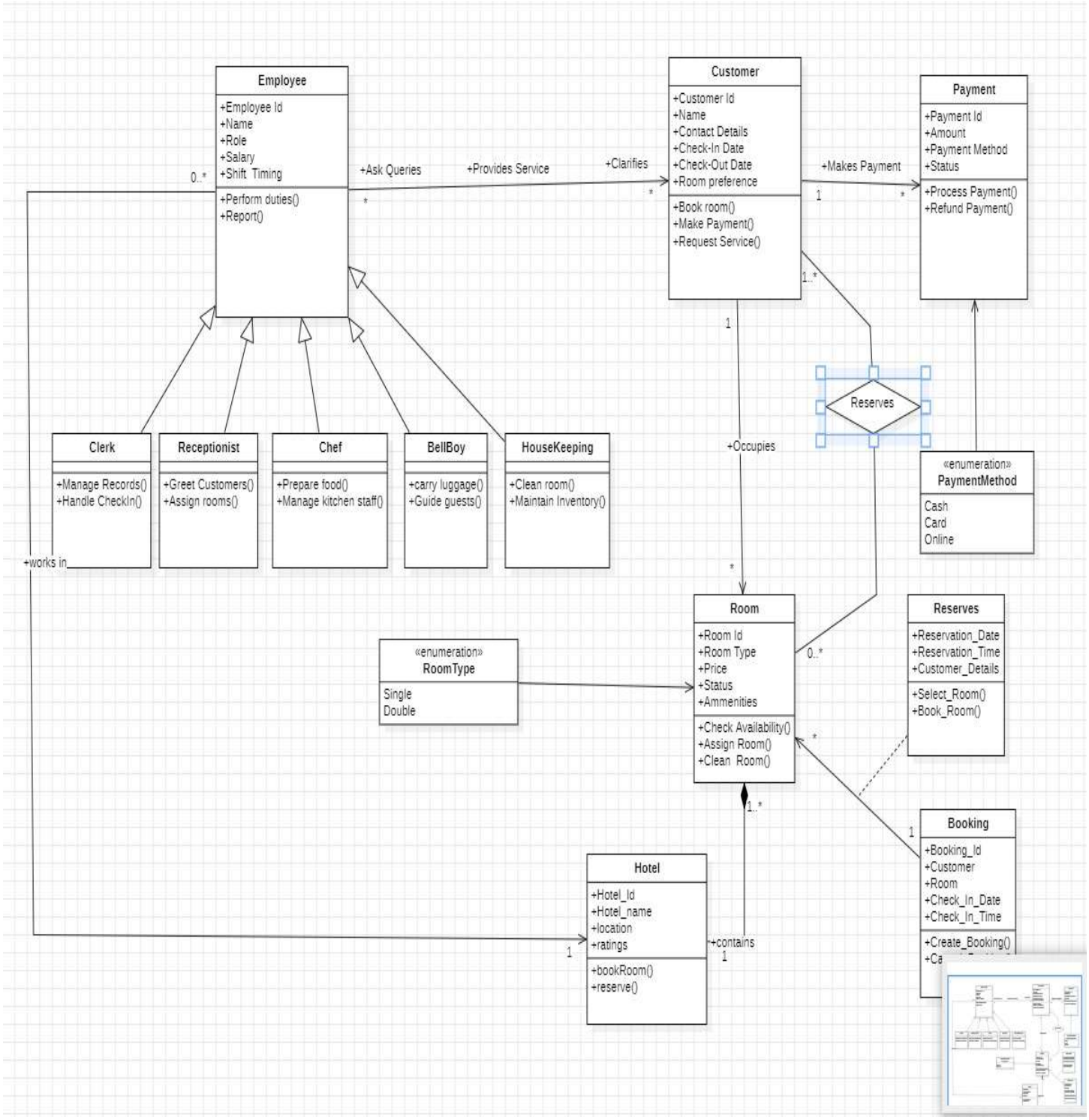
- **Schedule:**
    - **Phase 1:** Requirement Gathering (1 month)
    - **Phase 2:** System Design (1 month)

- **Phase 3:** Development (3 months)

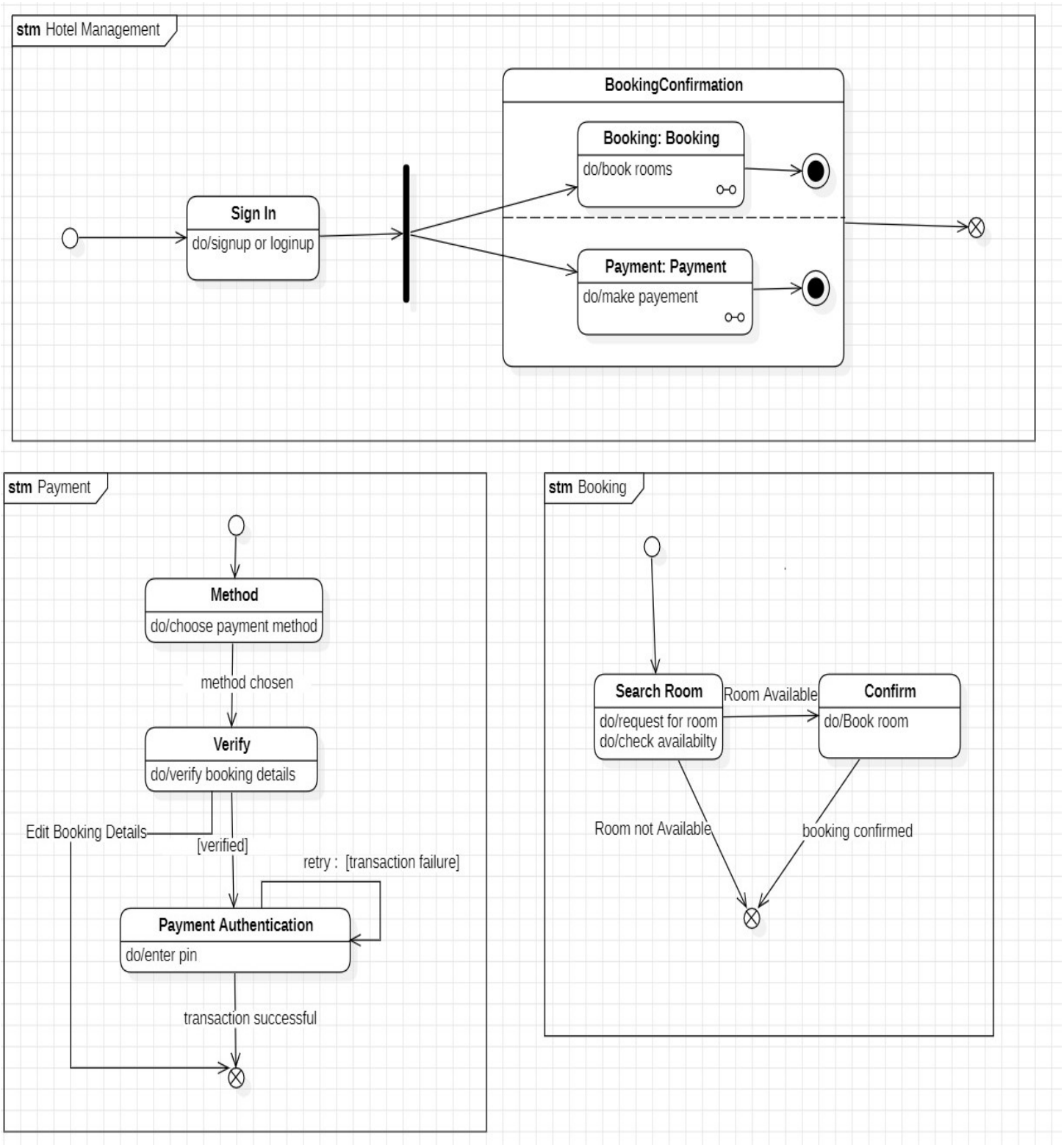
- **Phase 4:** Testing and Deployment (1 month)

- **Budget:**
  - **Development Cost:** $30,000
  - **Hardware and Software:** $10,000
  - **Testing and Deployment:** $5,000
  - **Miscellaneous:** $5,000

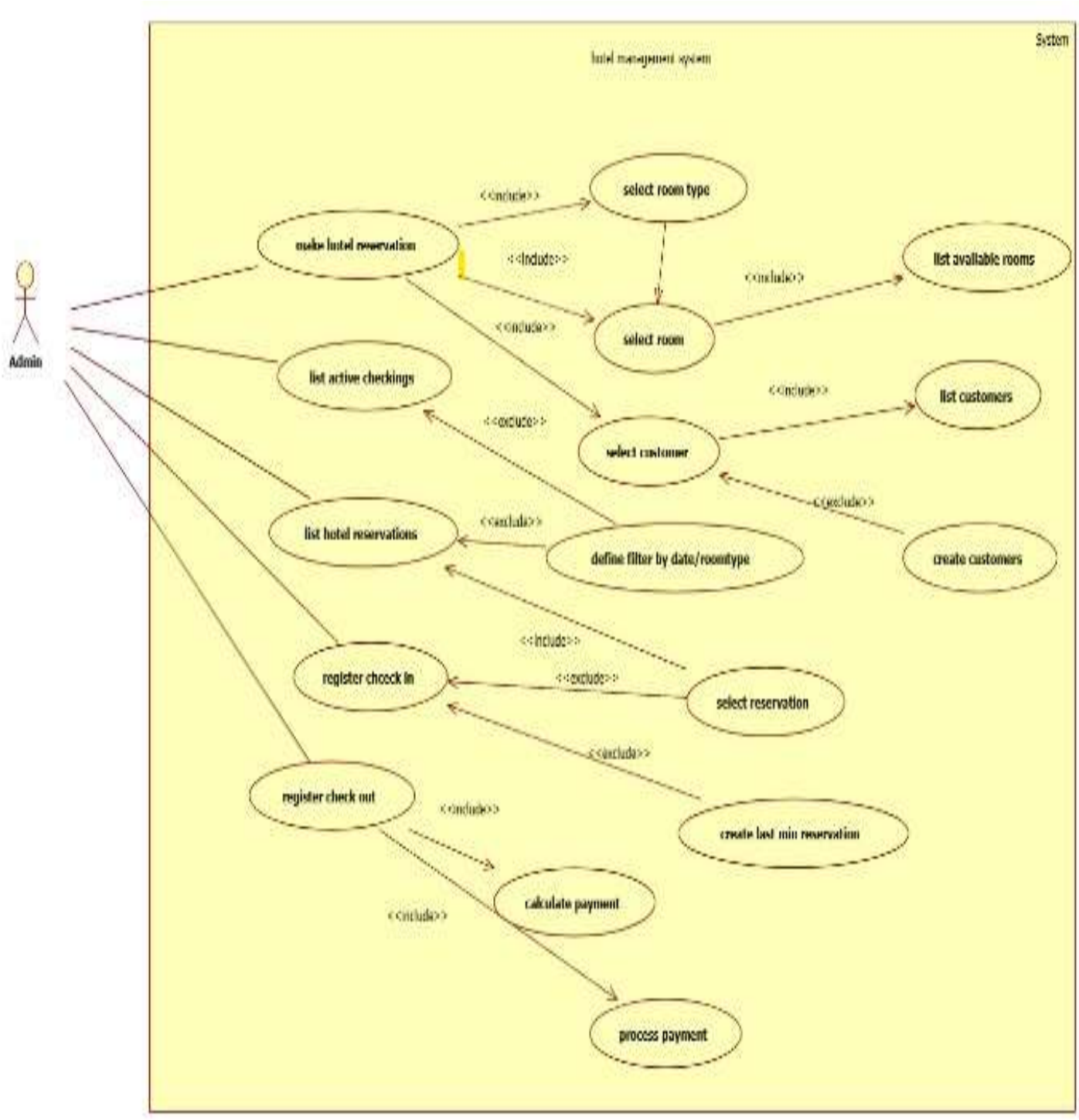
Advanced Class Diagram



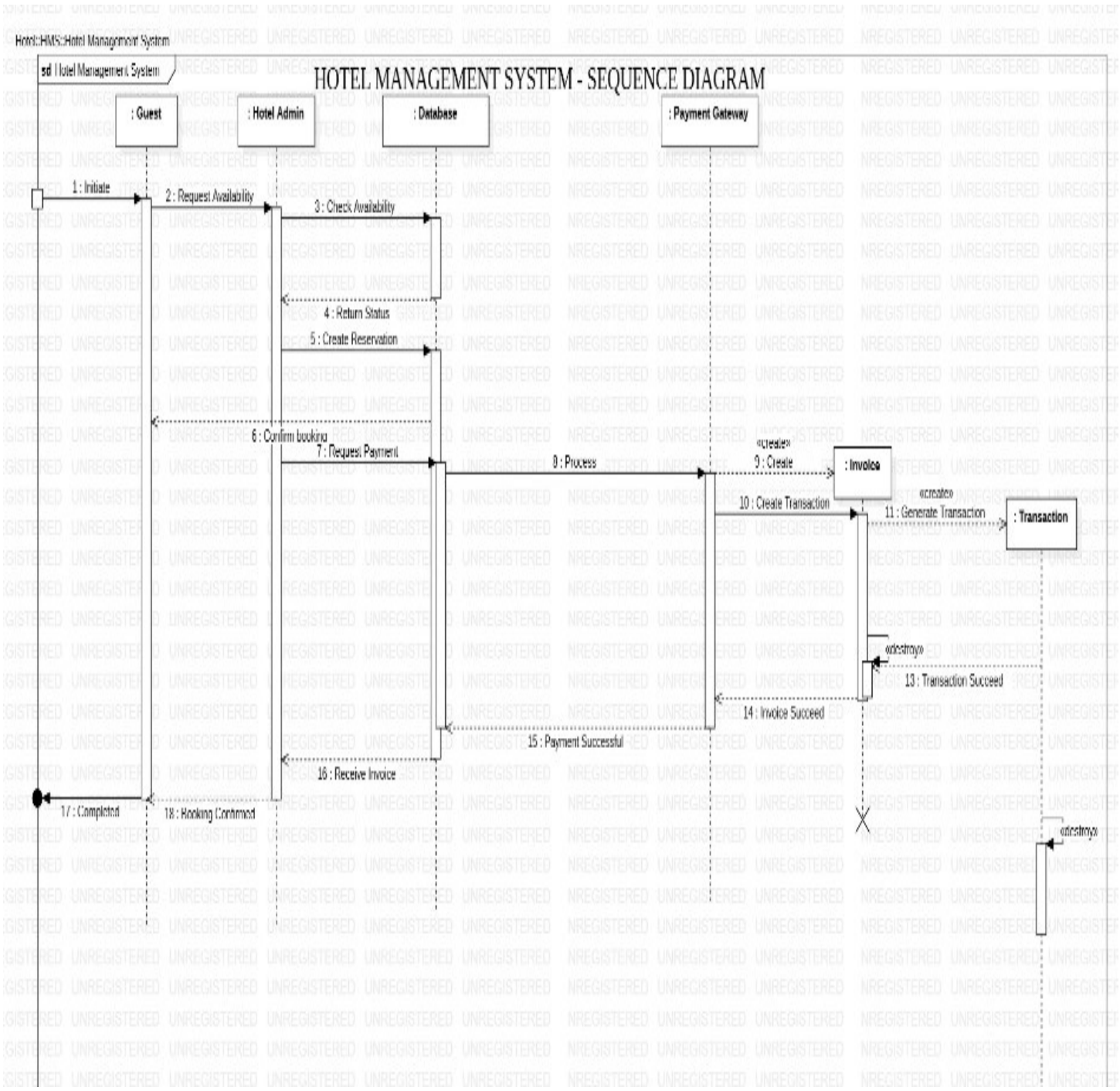
Advanced State Diagram



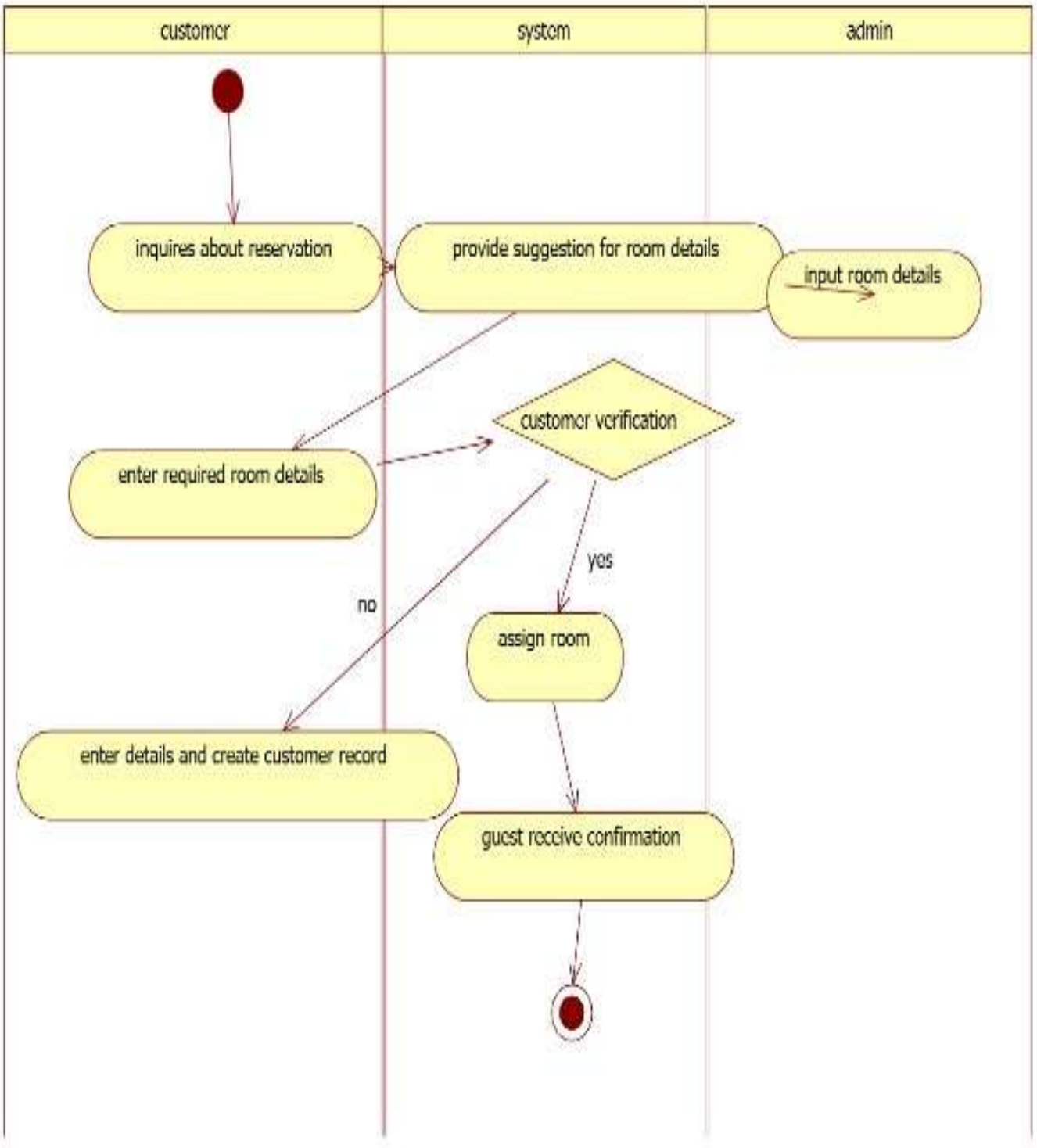
Advanced Use Case



Advanced Sequence Diagram



Advanced Activity Diagram



2. Credit Card Processing

Problem statement

The credit card processing system facilitates secure and efficient handling of credit card transactions. It ensures accurate validation, authorization, and settlement of payments. The system must include fraud detection mechanisms, encryption for data security, and integration with multiple banks, enabling seamless transaction processing for both customers and merchants.

SRS Document

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to define the requirements for a Credit Card Management System (CCMS). This document provides a clear understanding of the system's objectives, functionalities, constraints, and scope for developers, stakeholders, and testers to ensure efficient system development and implementation.

1.2 Scope of the Document

The Credit Card Management System will provide users with secure credit card management functionalities, including card applications, transactions, payments, statements, and fraud detection. The system will enhance the efficiency of card management processes for financial institutions and improve customer satisfaction. The development is estimated to require eight months with a budget of $80,000.

1.3 Overview

The CCMS is an integrated platform designed to manage credit card operations securely. It enables users to apply for cards, manage accounts, view transactions, make payments, and receive notifications. Financial institutions can use the system to monitor card activity, detect fraud, and generate detailed reports. This system will improve the reliability, efficiency, and transparency of credit card operations.

2. General Description

The system aims to:

- Provide a seamless credit card application and management process for users.

- Enable financial institutions to efficiently manage cardholder data, transactions, and risks.
- Enhance security with advanced fraud detection mechanisms.

Features and Benefits:

- Simplified credit card application and approval process.
- Real-time transaction tracking for cardholders.
- Automated billing and payment reminders.
- Fraud detection and prevention system to safeguard transactions.

- Comprehensive reporting for card issuers.

User Characteristics:

- **Cardholders:** Access accounts to manage transactions, payments, and statements.
- **Bank Staff:** Manage card applications, transaction disputes, and customer support.
- **Admins:** Oversee the system, manage user roles, and generate financial reports.

3. Functional Requirements

- **Card Application and Approval:** Users can apply for credit cards online, and bank staff can approve or reject applications based on eligibility.
- **Account Management:** Cardholders can view card details, manage limits, and block/unblock cards.
- **Transaction Monitoring:** Real-time display of transactions with detailed breakdowns.
- **Billing and Payments:** Generate statements and allow users to pay bills securely online.
- **Fraud Detection:** Monitor transactions for anomalies and alert users/banks of suspicious activities.
- **Notifications:** Send SMS and email alerts for transactions, payments, and suspicious activities.
- **Reporting:** Generate reports on transaction volumes, cardholder activities, and system performance.

4. Interface Requirements

- **User Interface:** A secure and intuitive web and mobile application for cardholders and bank staff.

- **Database Interface:** Integration with a robust database for storing sensitive user information and transaction logs.
- **Payment Gateway Interface:** Integration with secure payment systems for processing credit card payments.
- **Fraud Detection API:** Integration with third-party tools or algorithms for fraud detection and prevention.

5. Performance Requirements

- Support up to 1,000 simultaneous users with efficient performance.
- Handle up to 10,000 transactions per minute during peak usage.
- Fraud detection system must identify suspicious activities within 5 seconds of a transaction.
- Ensure 24/7 system availability with 99.99% uptime.

6. Design Constraints

- Must comply with regulatory standards like PCI DSS and GDPR for data security.
- Ensure compatibility with all major operating systems and web browsers.
- Use industry-standard encryption for data transmission and storage.
- Incorporate AI-based algorithms for fraud detection.

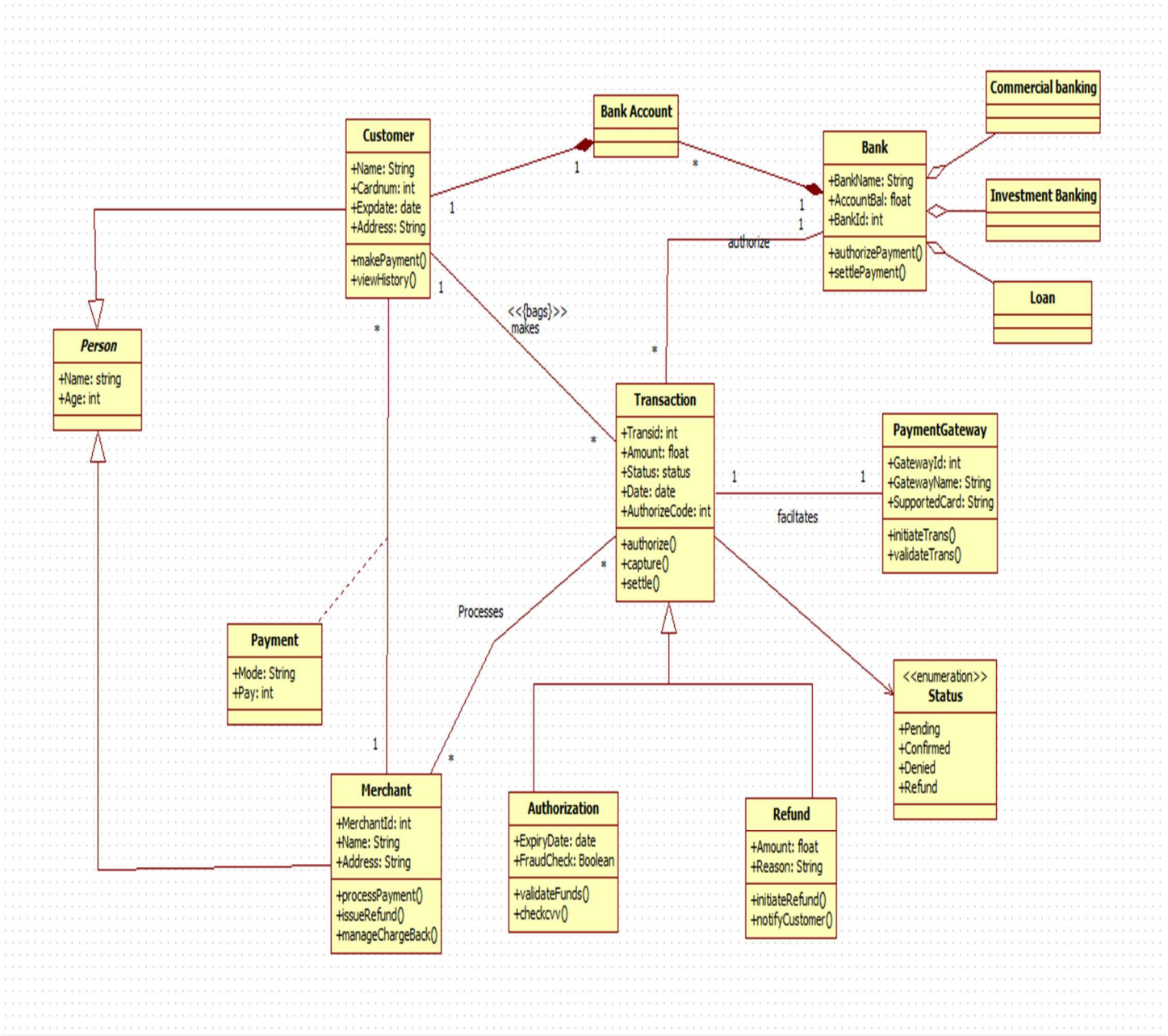
7. Non-Functional Attributes

- **Security:** Use advanced encryption protocols and multi-factor authentication.
- **Scalability:** Support growing user and transaction volumes as the system scales.
- **Reliability:** Maintain a maximum downtime of 1 hour per year.
- **Performance:** Ensure seamless operation even during peak transaction periods.
- **Portability:** Provide support for web and mobile platforms (iOS, Android).

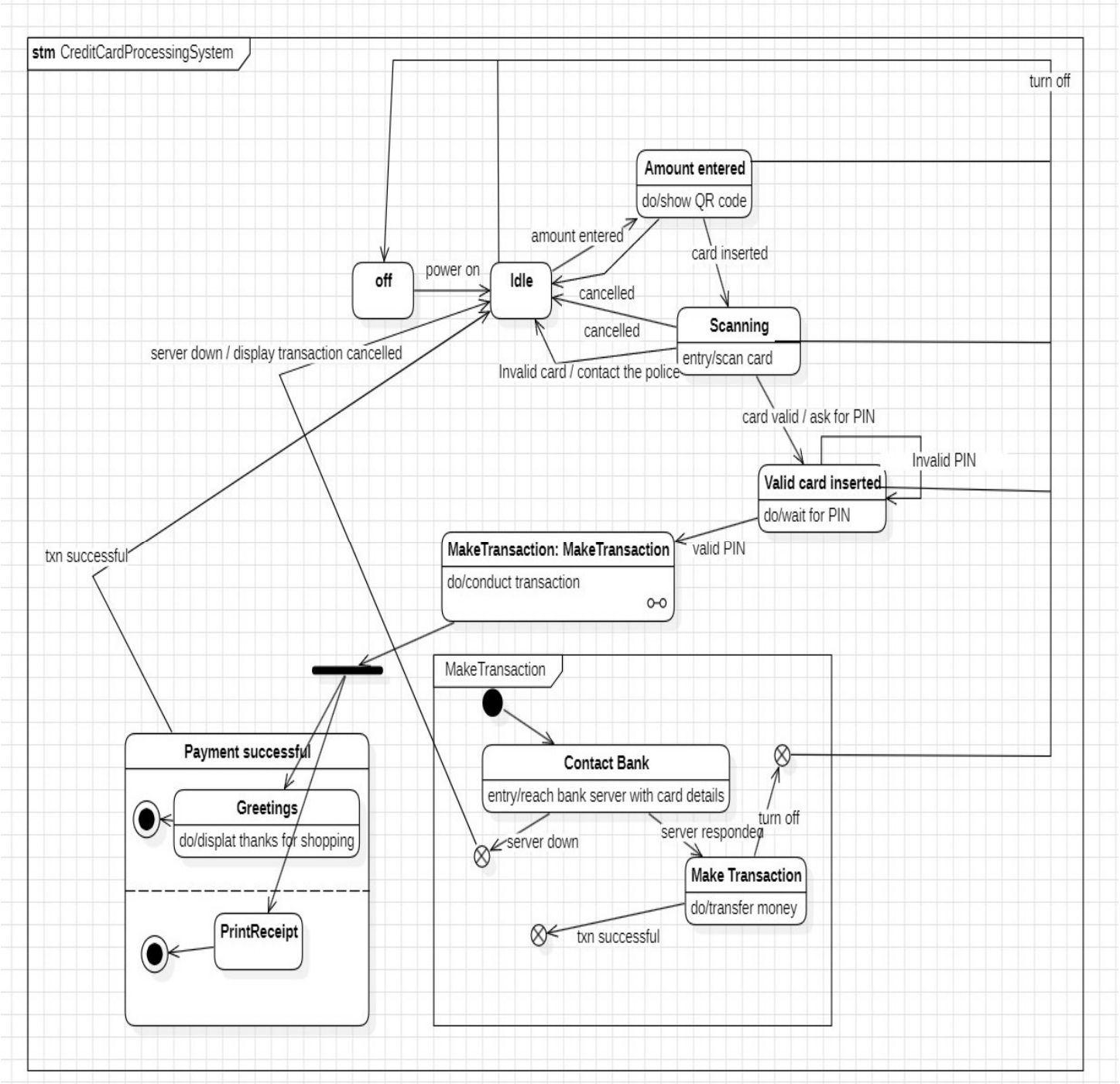
8. Preliminary Schedule and Budget

- **Schedule:**
    - **Phase 1:** Requirement Analysis (1 month)
    - **Phase 2:** Design and Prototyping (2 months)
    - **Phase 3:** Development (4 months)
    - **Phase 4:** Testing and Deployment (1 month)
- **Budget:**
    - **Development Cost:** $50,000
    - **Fraud Detection Tools:** $15,000
    - **Testing and Deployment:** $10,000

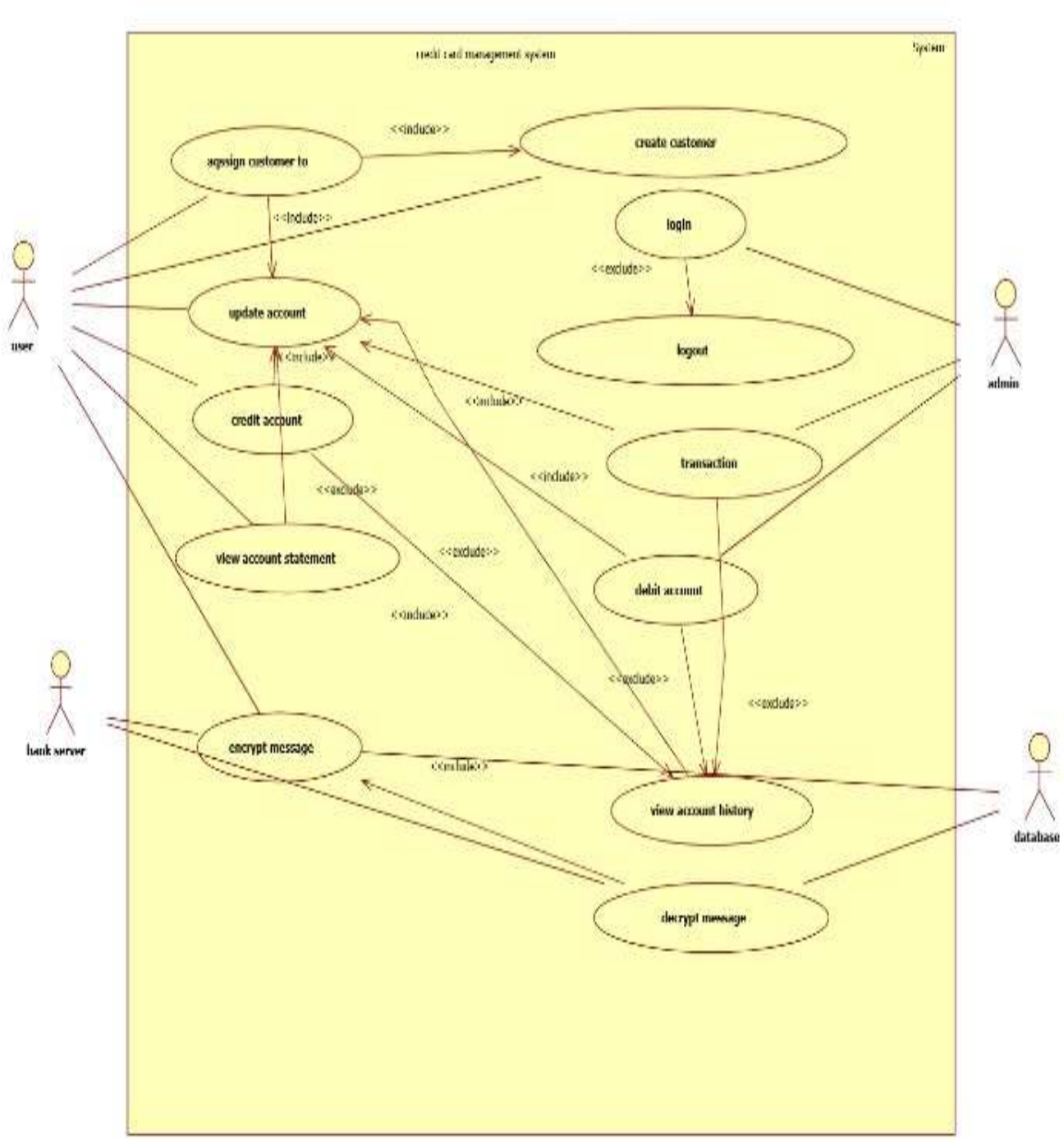
Advanced Class Diagram



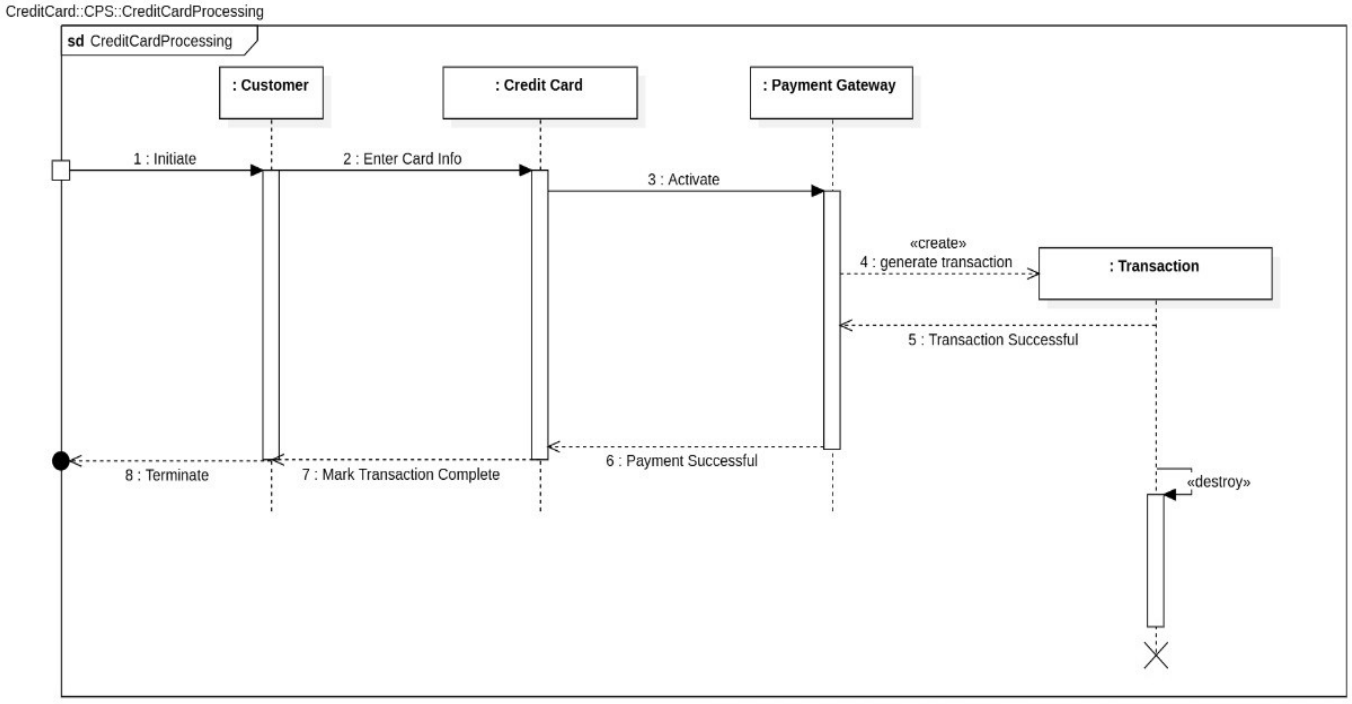
Advanced State Diagram



Advanced Use Case

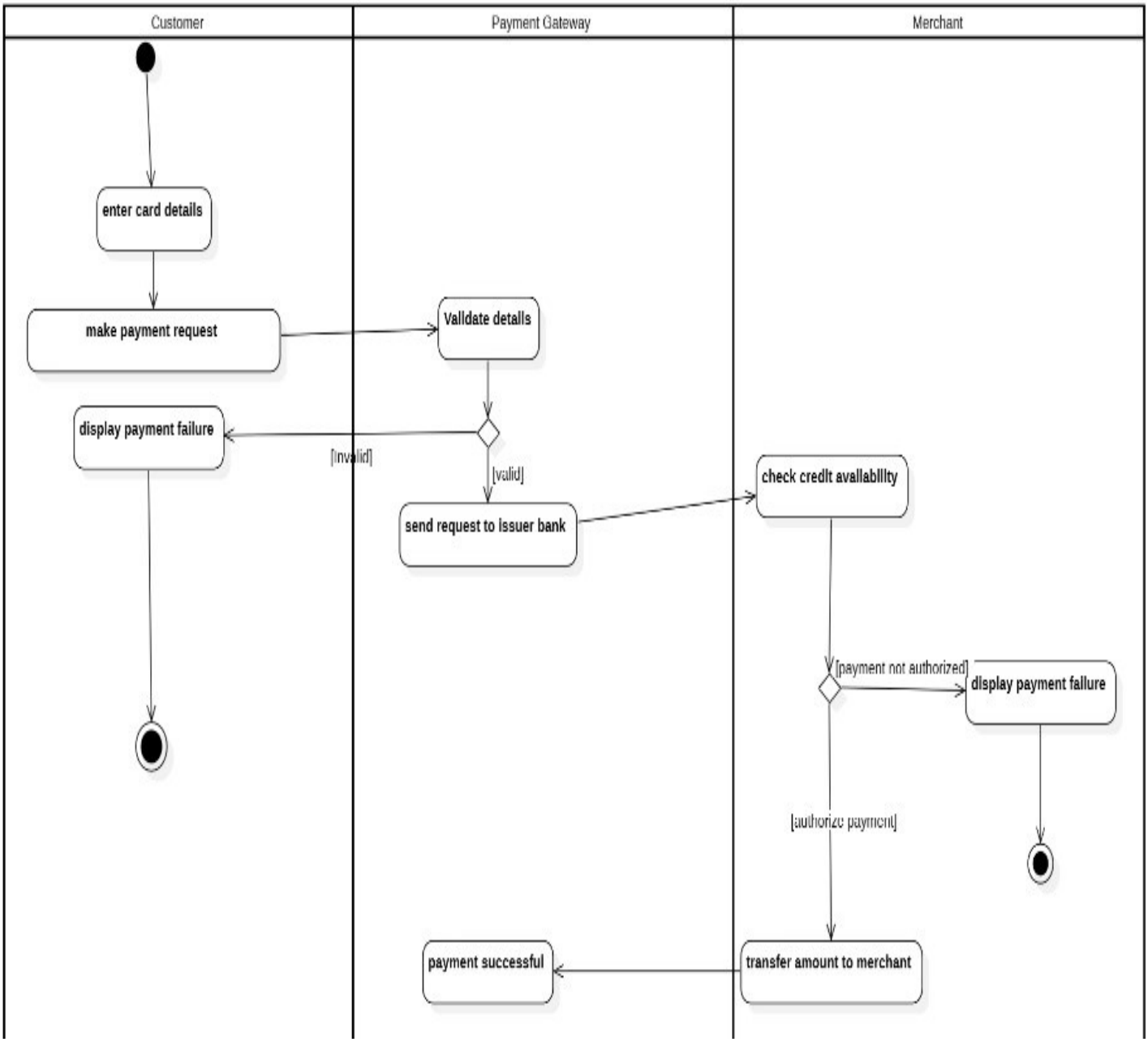


Advanced Sequence Diagram



CREDIT CARD PROCESSING SYSTEM - ADV SEQUENCE DIAGRAM

Advanced Activity Diagram



3.Library Management System

Problem statement

The library management system is designed to automate the operations of a library. It manages book records, borrowing, and returning processes, as well as user accounts. The system ensures efficient tracking of books, late fee calculation, and generates reports on book inventory and user activity. This reduces manual errors and enhances user convenience.

SRS Document

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to outline the requirements for a Library Management System (LMS). It serves as a reference for developers, stakeholders, and testers to ensure the development of a system that simplifies library operations, enhances user experience, and provides robust management tools for administrators.

1.2 Scope of the Document

The Library Management System will automate library processes, including cataloging, borrowing, returning, and inventory management. It will provide users (students, faculty, and librarians) with a seamless experience for managing books, records, and other library resources. The project is expected to take six months to complete with a development cost of approximately $40,000.

1.3 Overview

The LMS is a web-based application designed to streamline library operations. It offers user-friendly interfaces for library staff and patrons, enabling efficient management of books, periodicals, and digital resources. Key functionalities include search and reserve, borrowing and returning, fines management, and reporting.

2. General Description

The system aims to:

- Automate library operations to reduce manual errors and save time.
- Provide a user-friendly platform for patrons to access and manage library resources.

- Enhance administrative efficiency in tracking and managing inventory and memberships.

Features and Benefits:

- Centralized catalog for easy search and access to library resources.
- Automated borrowing, returning, and fine calculation processes.
- Integration with digital resource platforms for e-books and journals.
- Detailed reporting and analytics for library administrators.

User Characteristics:

- **Patrons (Students and Faculty):** Can search, reserve, borrow, and return books.
- **Librarians:** Manage inventory, process borrowing/returning, and monitor fines.
- **Admins:** Oversee system operations, user roles, and generate usage reports.

3. Functional Requirements

- **Search and Cataloging:** Users can search for books by title, author, or ISBN. Librarians can add or update catalog entries.
- **Borrowing and Returning:** Patrons can borrow and return books with automated due date tracking.
- **Reservation System:** Users can reserve unavailable books and get notified when they are ready for pickup.
- **Fines and Penalties:** Automatic fine calculation for overdue books and integration with online payment gateways.
- **Inventory Management:** Track stock, damaged books, and acquisition of new materials.
- **User Management:** Manage member registrations and roles (student, faculty, librarian).
- **Reports:** Generate detailed reports on borrowing trends, inventory usage, and fines collected.

4. Interface Requirements

- **User Interface:** Web-based and mobile-friendly interface for patrons and staff.
- **Database Interface:** Integration with a secure database to store book and user information.

- **Payment Gateway Interface:** Integration for fine payments via third-party systems like PayPal or Razorpay.
- **Digital Resource Integration:** Connect with external platforms for accessing e-books and journals.

5. Performance Requirements

- The system should support up to 500 simultaneous users.
- Search results must be displayed within 2 seconds.
- The system must handle up to 10,000 books in the catalog.
- Ensure 24/7 availability with a 99.9% uptime guarantee.

6. Design Constraints

- Compliance with library standards such as MARC (Machine-Readable Cataloging) for book metadata.
- Use of open-source frameworks to minimize development costs.
- Ensure compatibility with all major browsers and devices.
- Integration with RFID systems for book tracking.

7. Non-Functional Attributes

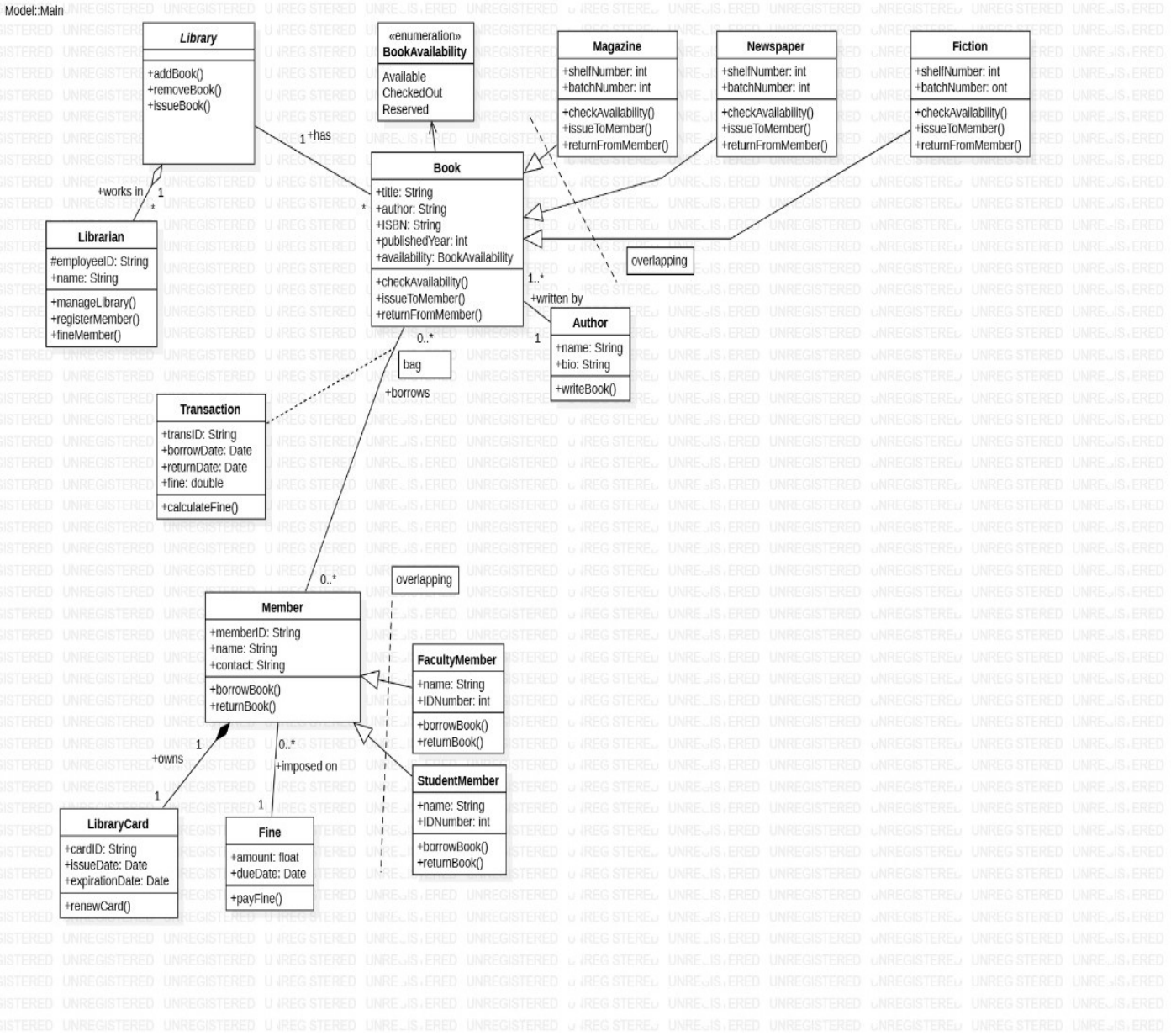
- **Security:** Protect user data with encryption and secure login mechanisms.
- **Scalability:** Support future library expansions and new functionalities.
- **Reliability:** Maintain a maximum downtime of 1 hour per month.
- **Performance:** Maintain fast response times for search and transaction processes.
- **Usability:** Provide a simple and intuitive interface for non-technical users.

8. Preliminary Schedule and Budget

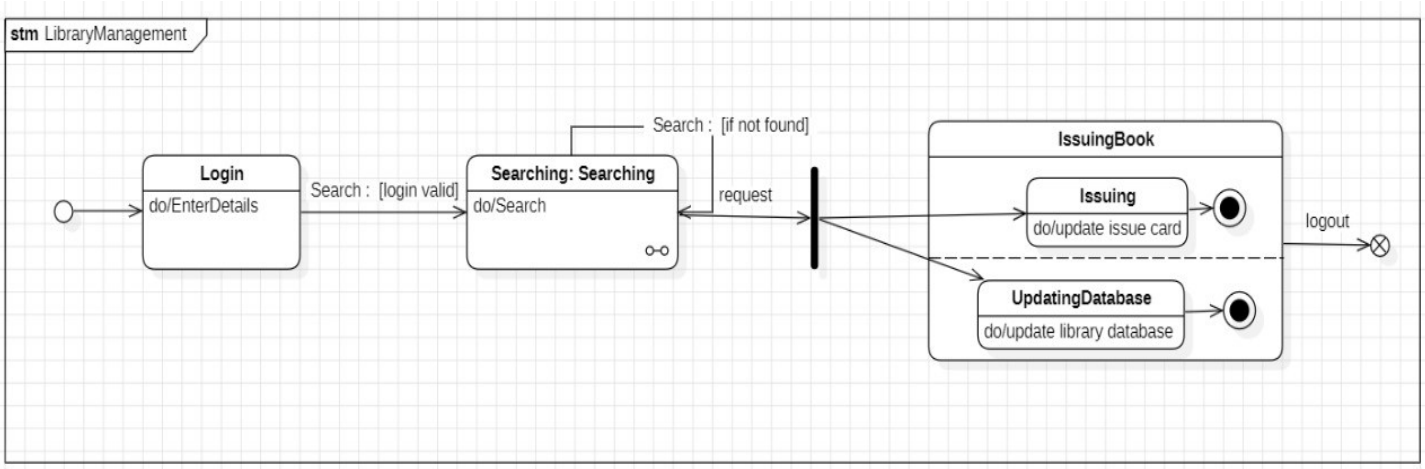
- **Schedule:**
    - **Phase 1:** Requirement Gathering (1 month)
    - **Phase 2:** System Design (1 month)

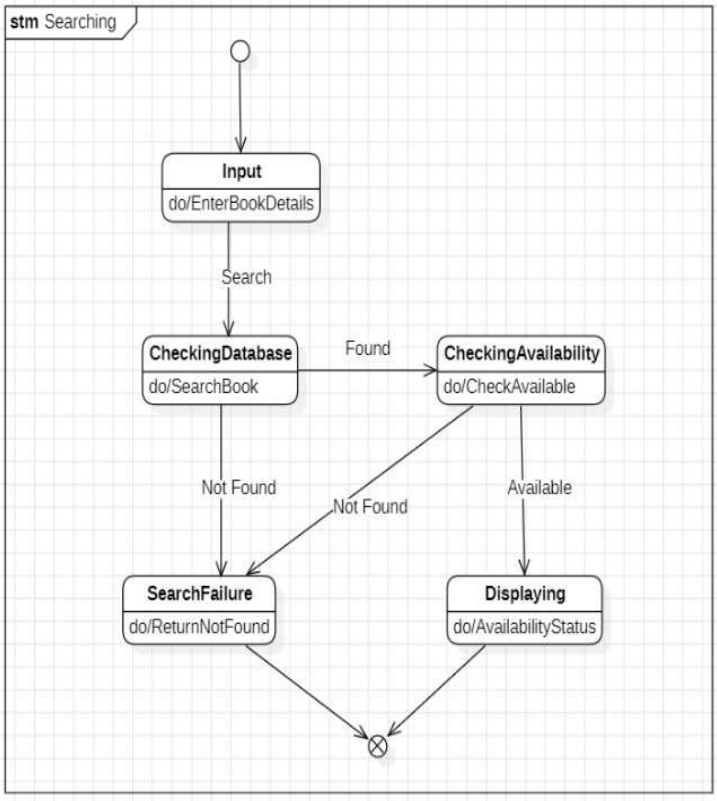
- **Phase 3:** Development (3 months)
- **Phase 4:** Testing and Deployment (1 month)

- **Budget:**
  - **Development Cost:** $25,000
  - **Hardware and Software:** $8,000
  - **Testing and Deployment:** $5,000
  - **Miscellaneous:** $2,000

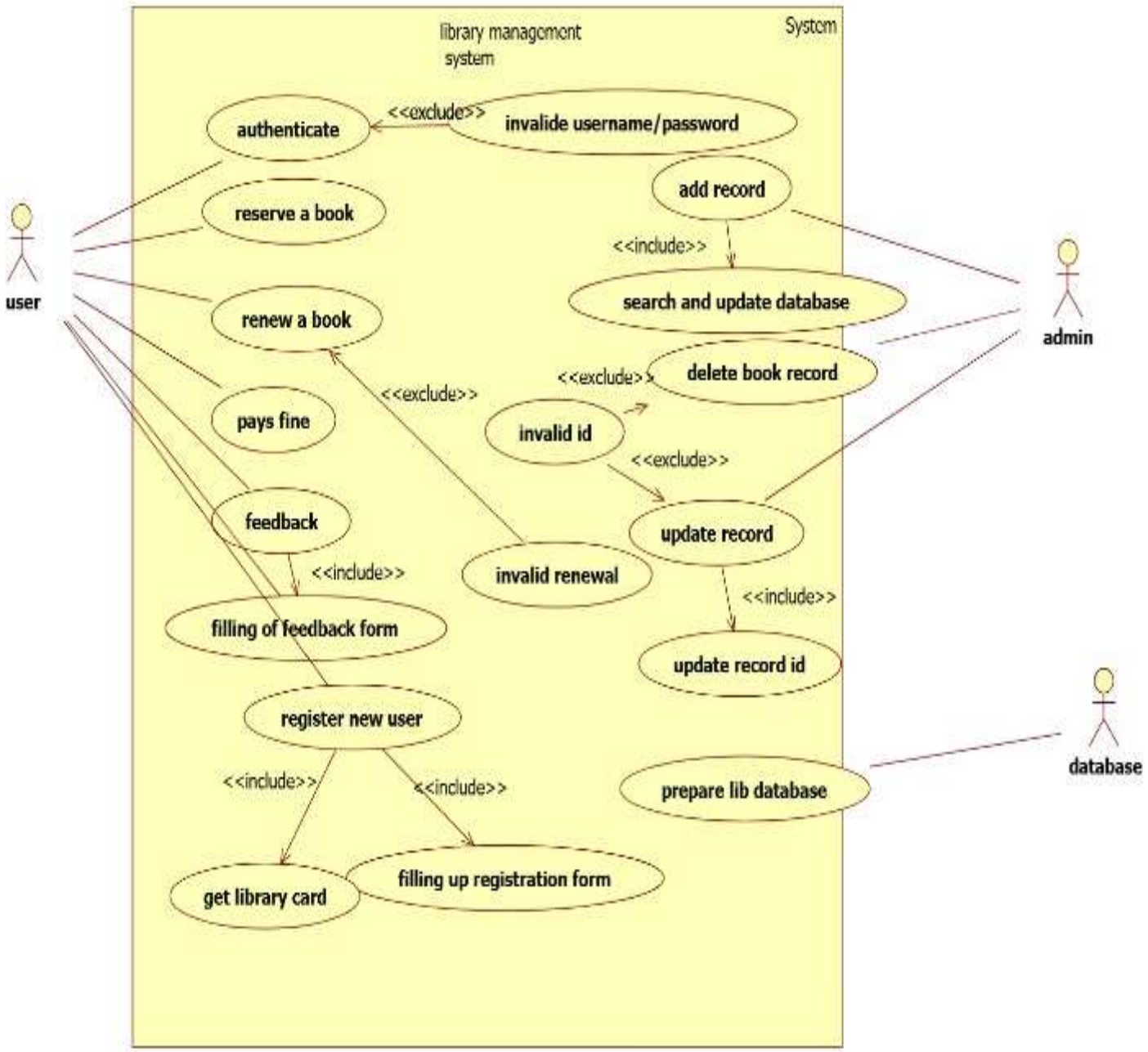

Model::Main
Library
+addBook()
+removeBook()
+issueBook()
«enumeration»
BookAvailability
Available
CheckedOut
Reserved
Magazine
+shelfNumber: int
+batchNumber: int
+checkAvailability()
+issueToMember()
+returnFromMember()
Newspaper
+shelfNumber: int
+batchNumber: int
+checkAvailability()
+issueToMember()
+returnFromMember()
Fiction
+shelfNumber: int
+batchNumber: ont
+checkAvailability()
+issueToMember()
+returnFromMember()
1 +has
+works in
1
*
Librarian
#employeeID: String
+name: String
+manageLibrary()
+registerMember()
+fineMember()
Book
+title: String
+author: String
+ISBN: String
+publishedYear: int
+availability: BookAvailability
+checkAvailability()
+issueToMember()
+returnFromMember()
overlapping
1..*
+written by
1
Author
+name: String
+bio: String
+writeBook()
0..*
bag
+borrows
Transaction
+transID: String
+borrowDate: Date
+returnDate: Date
+fine: double
+calculateFine()
0..*
overlapping
Member
+memberID: String
+name: String
+contact: String
+borrowBook()
+returnBook()
FacultyMember
+name: String
+IDNumber: int
+borrowBook()
+returnBook()
StudentMember
+name: String
+IDNumber: int
+borrowBook()
+returnBook()
1
+owns
1
0..*
+imposed on
1
LibraryCard
+cardID: String
+issueDate: Date
+expirationDate: Date
+renewCard()
Fine
+amount: float
+dueDate: Date
+payFine()

Advanced State Diagram

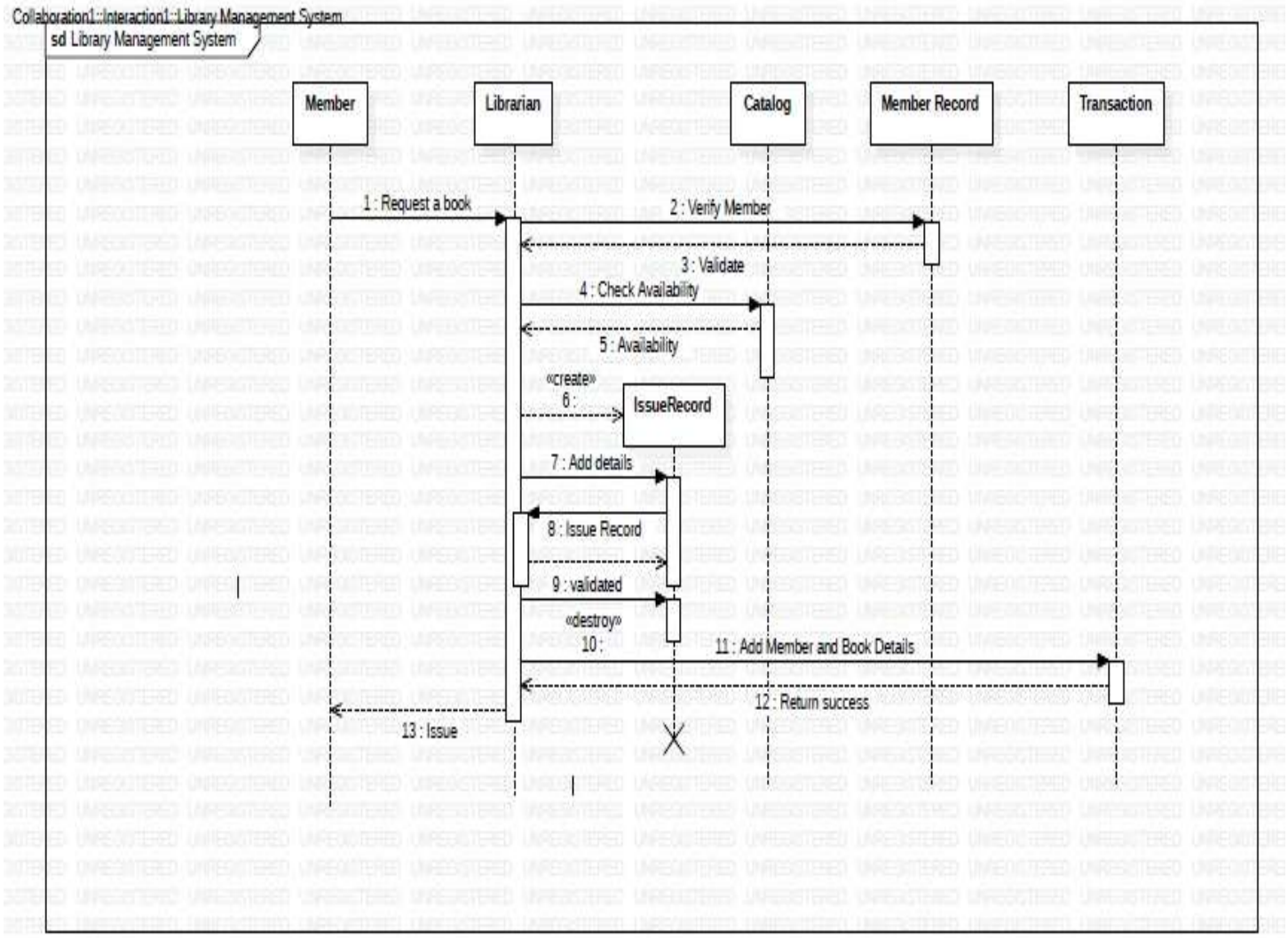




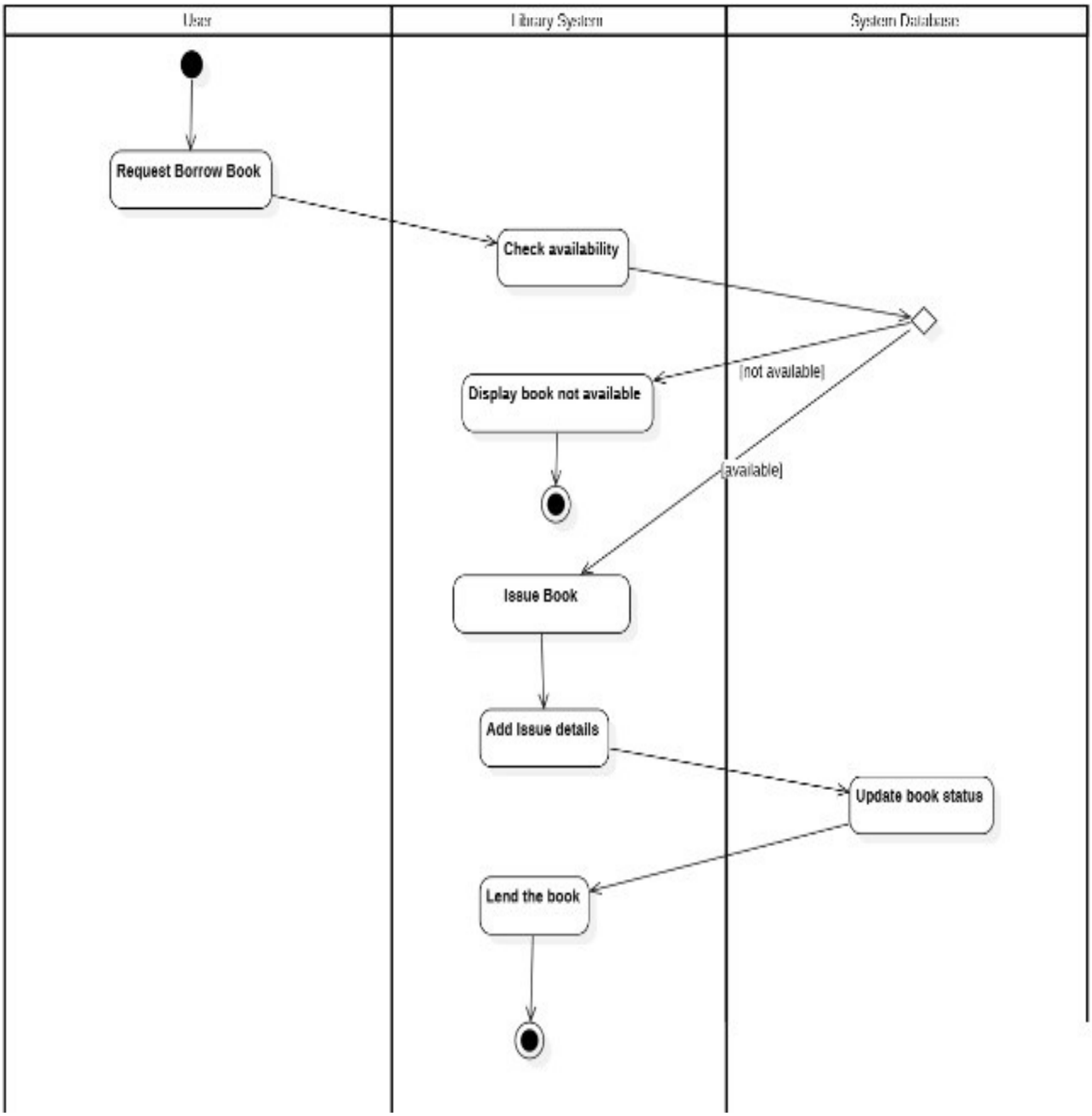
Advanced Use Case Diagram



Advanced Sequence Diagram



Advanced Activity Diagram



4.Stock Maintenance System

Problem statement

The stock maintenance system is designed to manage inventory effectively, keeping track of available stock, restocking requirements, and stock usage. The system ensures real-time updates on inventory levels, generates alerts for low stock, and provides detailed reports for analysis. This enables businesses to avoid overstocking or stockouts and maintain an optimal inventory balance.

SRS Document

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to define the requirements for a Stock Maintenance System (SMS). It serves as a comprehensive guide for stakeholders, developers, and testers to ensure the system fulfills the needs of efficiently managing inventory, tracking stock levels, and generating reports to streamline stock management.

1.2 Scope of the Document

The Stock Maintenance System will help businesses manage inventory, track stock movement, automate reordering processes, and generate insightful reports. The system is designed for warehouses, retail stores, and manufacturing units to reduce manual errors and optimize stock management. The development is estimated to take six months with an approximate budget of $45,000.

1.3 Overview

The SMS is a web-based application designed to manage stock inventory in real-time. It provides tools to track stock levels, manage suppliers, process orders, and analyze stock performance. The system ensures the availability of stock, minimizes overstocking or understocking, and improves operational efficiency.

2. General Description

The system aims to:

- Automate stock tracking and management processes.
- Provide accurate real-time data on stock levels and movements.
- Enhance decision-making through detailed reporting and analytics.

Features and Benefits:

- Centralized stock tracking system to manage inventory across multiple locations.
- Automatic reorder alerts for low stock levels.
- Barcode and RFID integration for efficient stock movement tracking.
- Supplier and purchase order management for seamless restocking.
- Detailed reports on inventory turnover, stock aging, and order history.

User Characteristics:

- **Warehouse Staff:** Manage incoming and outgoing stock and update inventory records.
- **Store Managers:** Monitor stock levels, place orders, and review stock performance.
- **Admins:** Oversee the system, assign roles, and generate high-level reports.

3. Functional Requirements

- **Stock Tracking:** Real-time tracking of stock levels, including current quantity, stock in transit, and reserved stock.
- **Reordering System:** Automatic alerts and generation of purchase orders when stock falls below predefined thresholds.
- **Barcode and RFID Integration:** Enable quick and accurate entry and tracking of stock items.
- **Supplier Management:** Maintain supplier details, manage purchase orders, and track supplier performance.
- **Reporting:** Generate reports on stock usage, turnover rates, and stock aging to identify trends and inefficiencies.
- **User Management:** Assign roles and permissions to different user groups (warehouse staff, managers, admins).
- **Audit Logs:** Maintain logs of all stock transactions for transparency and accountability.

4. Interface Requirements

- **User Interface:** A responsive web application and optional mobile app for inventory management on the go.
- **Database Interface:** Integration with a secure database to store stock, supplier, and order details.
- **Integration APIs:** APIs for integration with ERP systems, e-commerce platforms, or other external tools.
- **Barcode/RFID Interface:** Integration with barcode scanners or RFID readers for efficient stock handling.

6. Performance Requirements

- The system should handle up to 1,000 simultaneous users.
- Updates to stock levels must reflect in the system within 2 seconds.
- Ensure 24/7 availability with a 99.9% uptime guarantee.

6. Design Constraints

- Use industry-standard technologies and open-source frameworks to reduce costs.
- Ensure compatibility with Windows, macOS, and mobile platforms.
- Adhere to data protection regulations, such as GDPR or CCPA, to safeguard user and stock data.

7. Non-Functional Attributes

- **Security:** Implement encryption for data storage and communication, and use role-based access control (RBAC).
- **Reliability:** Maintain a maximum downtime of 1 hour per month for system updates or maintenance.
- **Scalability:** Allow seamless addition of new warehouses, users, or stock items as business grows.
- **Usability:** Provide an intuitive interface that minimizes training requirements for staff.

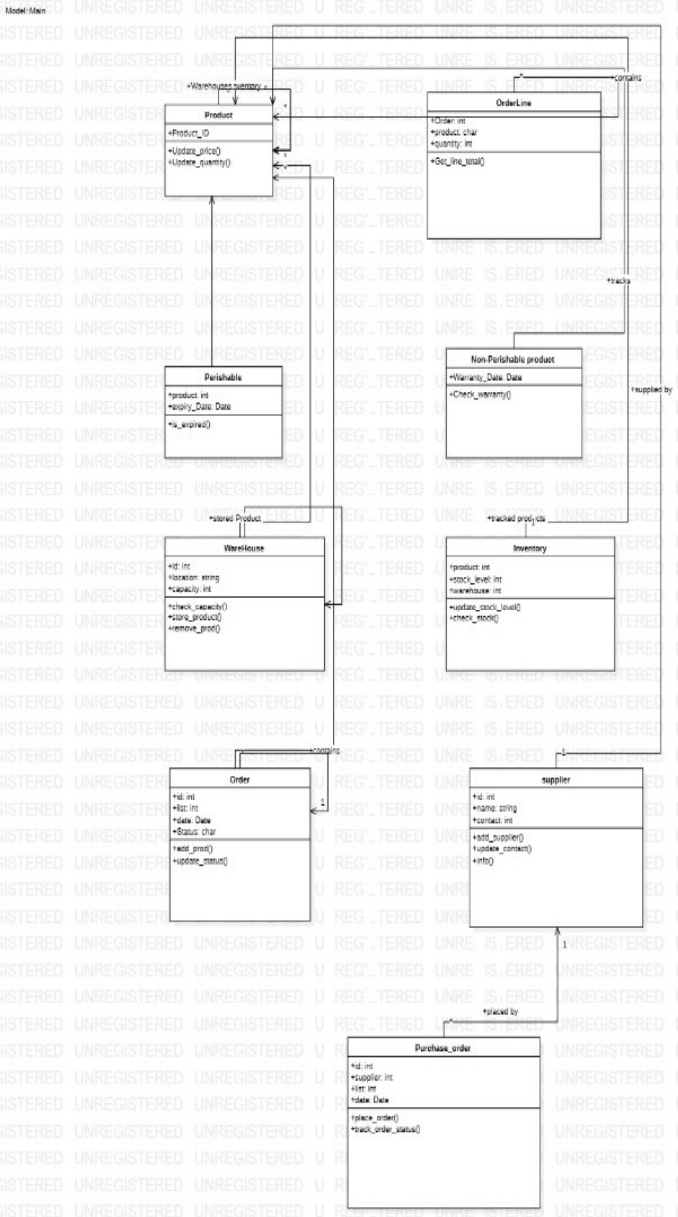
8. Preliminary Schedule and Budget

- **Schedule:**

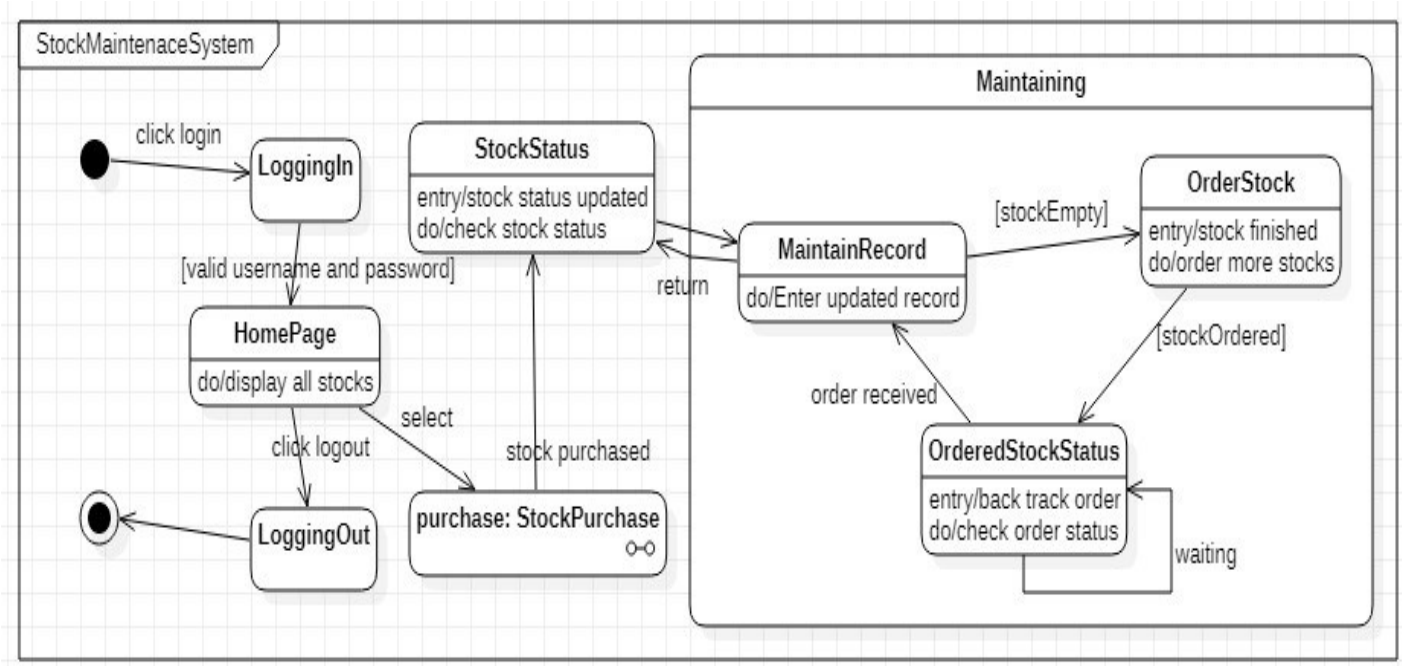
- **Phase 1:** Requirement Analysis (1 month)
- **Phase 2:** System Design (1 month)
- **Phase 3:** Development (3 months)
- **Phase 4:** Testing and Deployment (1 month)

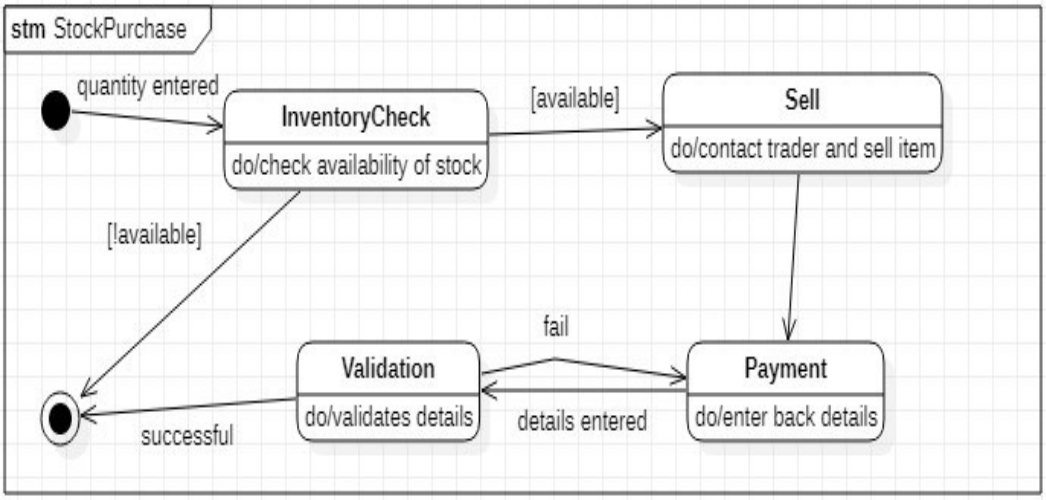
- **Budget:**
  - **Development Cost:** $30,000
  - **Hardware and Software:** $8,000
  - **Testing and Deployment:** $5,000

Advanced Class Diagram

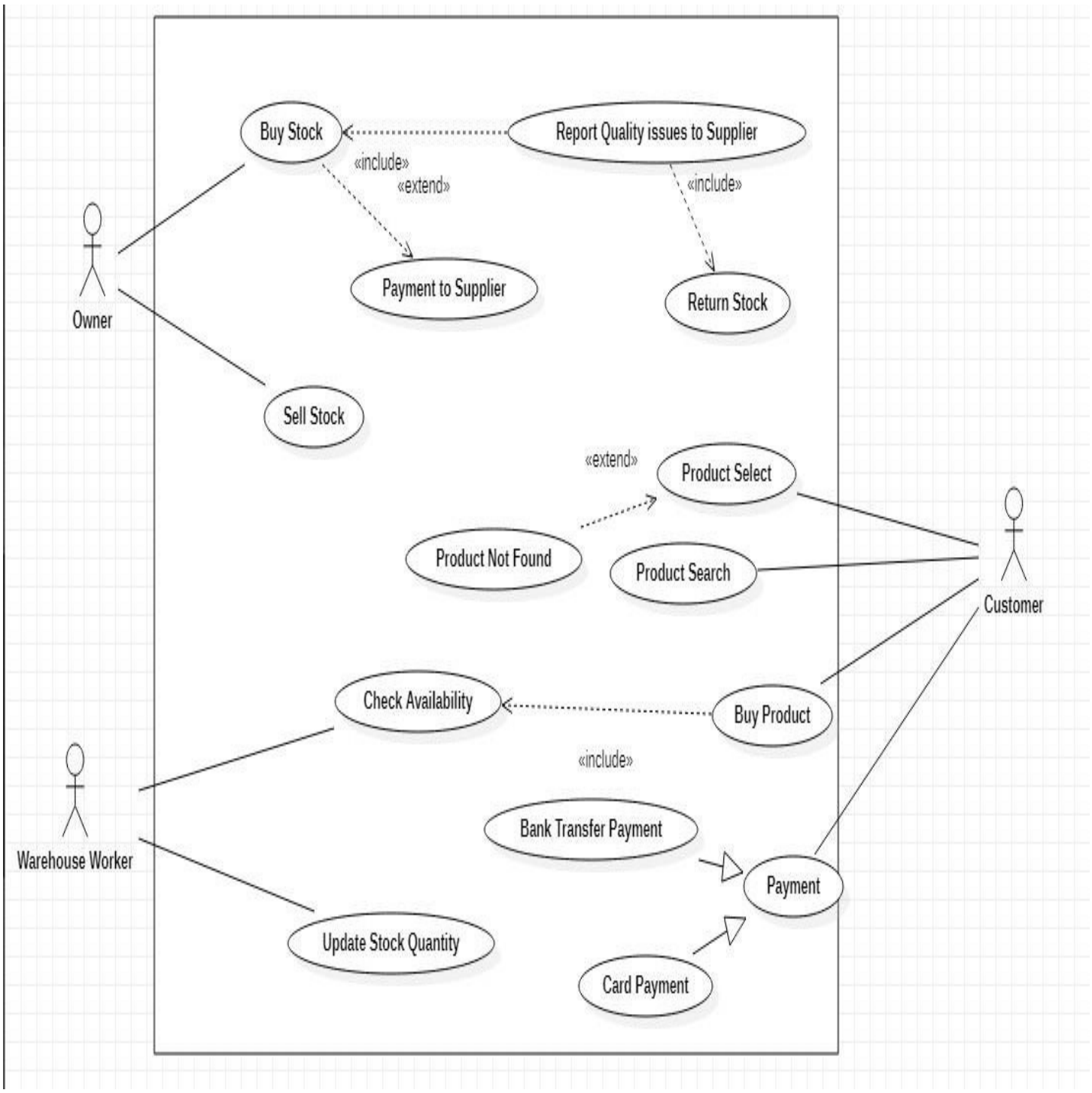


Advanced State Diagram

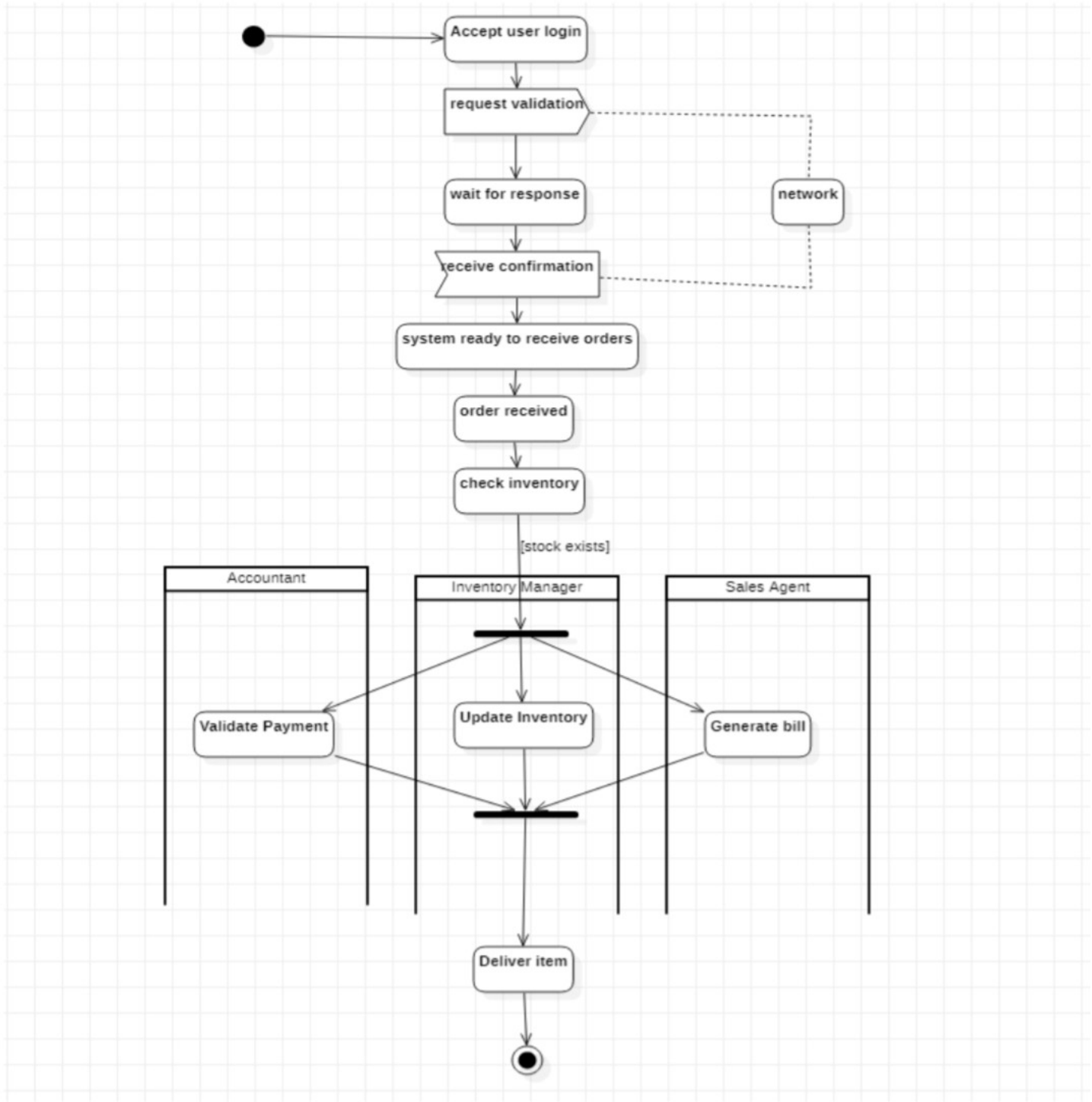




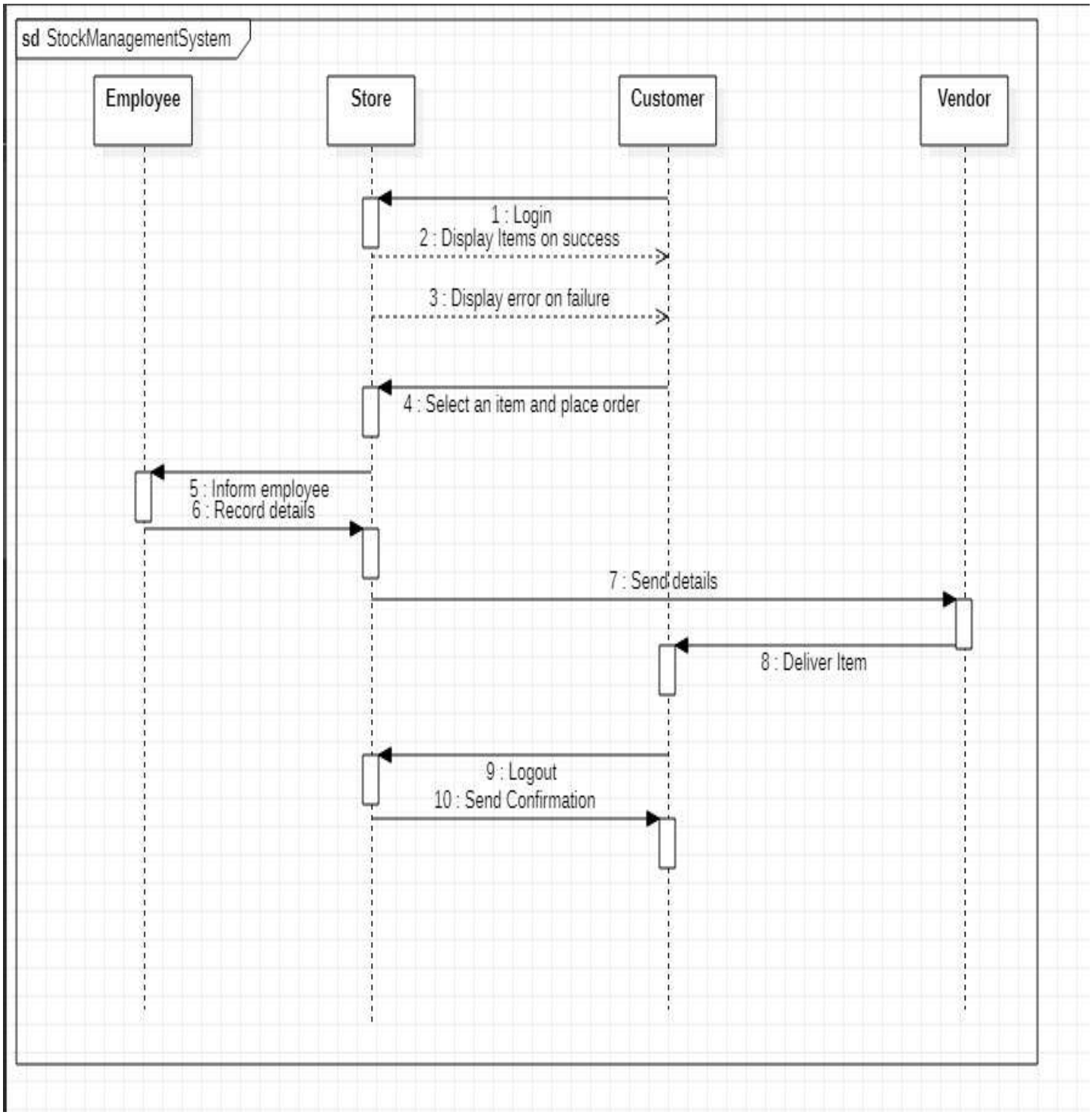
Advanced Use Case Diagram



Advanced Activity Diagram



Advanced Sequence Diagram



5. Passport Automation System

Problem statement

The passport automation system simplifies the process of applying for and issuing passports. It provides features for user registration, document submission, appointment scheduling, and tracking application status. The system aims to reduce manual interventions, ensure transparency, and enhance the user experience by making the entire process streamlined and time-efficient.

SRS Document

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to outline the requirements for developing a Passport Automation System (PAS). The document serves as a reference for stakeholders, developers, and testers to ensure that the system efficiently manages passport application, processing, and issuance while enhancing user experience and operational efficiency.

1.2 Scope of the Document

The Passport Automation System will streamline the process of applying for, tracking, and issuing passports. It aims to reduce manual intervention, eliminate errors, and enhance transparency in passport processing. The system will be designed for citizens, government officials, and passport offices. The project is estimated to take eight months to complete, with a budget of $60,000.

1.3 Overview

The PAS is a centralized, web-based application that automates the end-to-end process of passport management. It allows citizens to apply online, schedule appointments, upload required documents, track application status, and receive notifications. Government officials can process applications, verify documents, and issue passports efficiently. The system ensures data security and compliance with international standards.

2. General Description

The system aims to:

- Provide a user-friendly interface for citizens to apply for passports and track applications.

- Enable efficient management of passport applications, verifications, and issuance by government officials.
- Ensure secure storage and processing of sensitive personal data.

Features and Benefits:

- Online application submission and appointment scheduling.
- Document uploading and verification.
- Real-time application status tracking and notifications.
- Automated passport generation and dispatch tracking.
- Analytics and reporting for government use.

User Characteristics:

- **Citizens:** Apply for new passports, renewals, or updates.
- **Government Officials:** Process applications, verify documents, and issue passports.
- **Admins:** Manage the system, monitor performance, and generate high-level reports.

3. Functional Requirements

- **Application Submission:** Allow users to fill out application forms, upload documents, and pay fees online.
- **Appointment Scheduling:** Enable users to schedule appointments at passport offices or verification centers.
- **Document Verification:** Government officials can review uploaded documents and mark applications as verified or rejected.
- **Status Tracking:** Allow users to track application progress in real-time.
- **Notification System:** Send email/SMS notifications for application updates, appointment reminders, and passport dispatch.
- **Passport Issuance:** Automatically generate and print passports once the application is approved.

- **Reporting and Analytics:** Generate reports on application trends, processing times, and system performance.

4. Interface Requirements

- **User Interface:** A web-based portal for citizens and a secure dashboard for government officials.
- **Database Interface:** Integration with a robust database to store application and user information securely.

- **Biometric System Integration:** Interfaces with biometric systems for identity verification (fingerprints, facial recognition).
- **Payment Gateway Interface:** Integration with secure payment systems for processing application fees.

5. Performance Requirements

- Handle up to 5,000 simultaneous users.
- Ensure that the system processes and updates application status within 5 seconds.
- Support the storage of up to 10 million user records securely.
- Maintain 24/7 availability with a 99.99% uptime guarantee.

6. Design Constraints

- Compliance with international standards for data protection, such as GDPR.
- Adherence to security protocols, including SSL encryption for all data transmissions.
- Use of scalable technologies to accommodate future increases in application volumes.
- Integration with government identity databases for automated verification.

7. Non-Functional Attributes

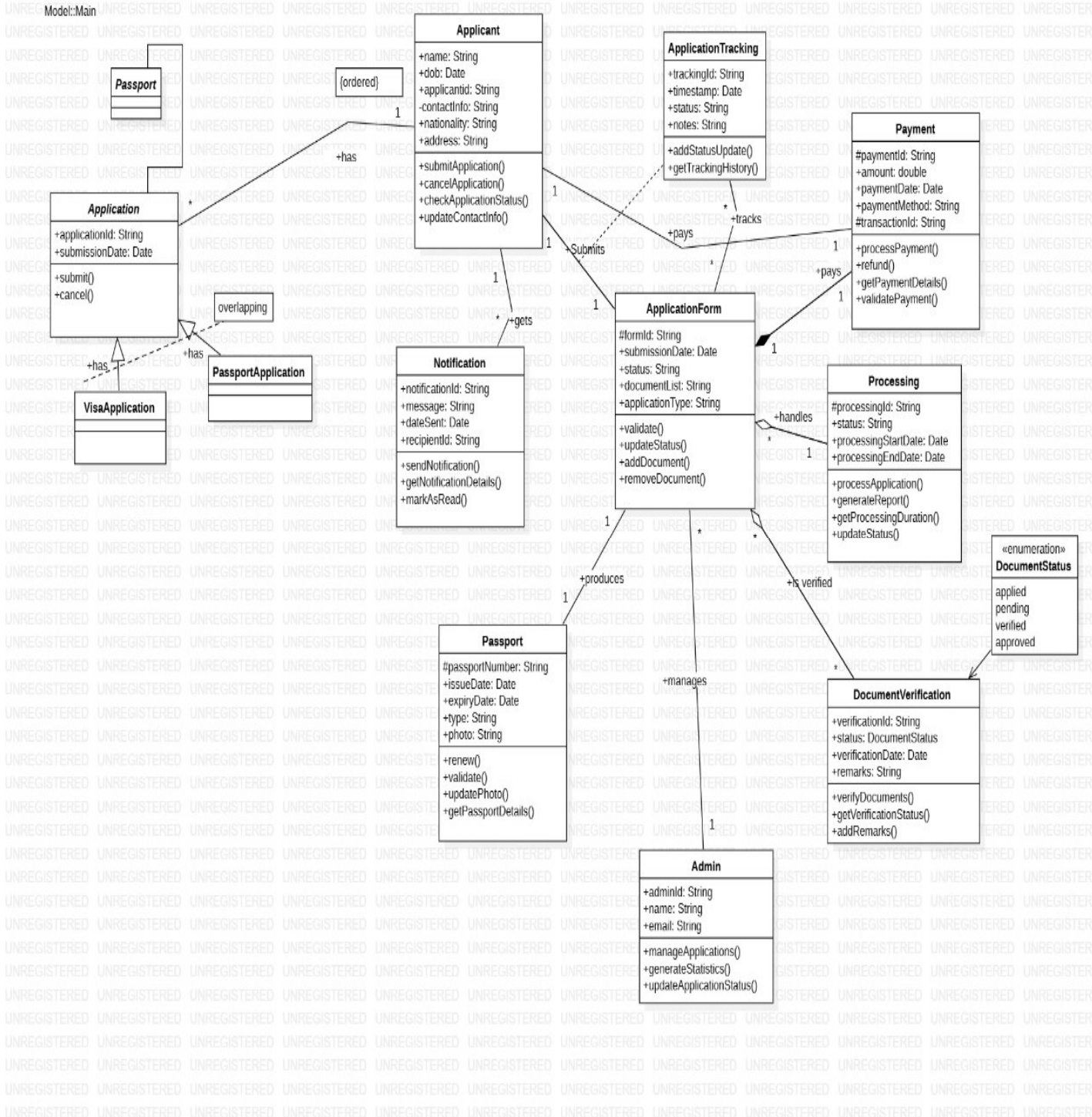
- **Security:** Employ advanced encryption for sensitive data and multi-factor authentication for system access.
- **Scalability:** Support the addition of new passport offices and increased user load.

- **Reliability:** Ensure a maximum downtime of 30 minutes per month for system maintenance.
- **Usability:** Provide an intuitive interface for users with minimal technical knowledge.
- **Portability:** Support access from desktops, tablets, and mobile devices.

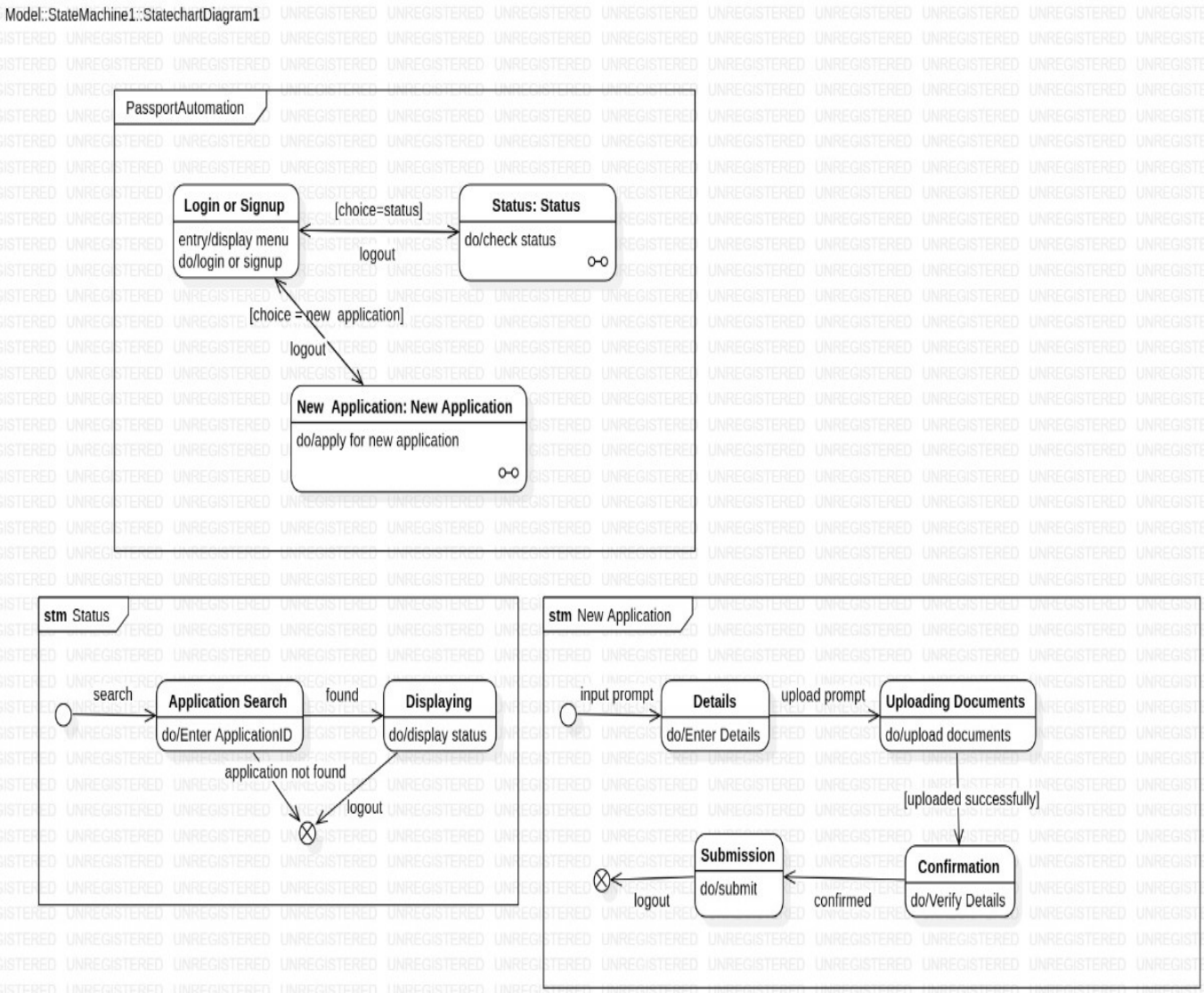
8. Preliminary Schedule and Budget

- **Schedule:**
    - **Phase 1:** Requirement Analysis (1 month)
    - **Phase 2:** System Design (2 months)
    - **Phase 3:** Development (3 months)
    - **Phase 4:** Testing and Deployment (2 months)
- **Budget:**
    - **Development Cost:** $40,000
    - **Hardware and Software:** $12,000
    - **Testing and Deployment:** $6,000
    - **Miscellaneous:** $2,000

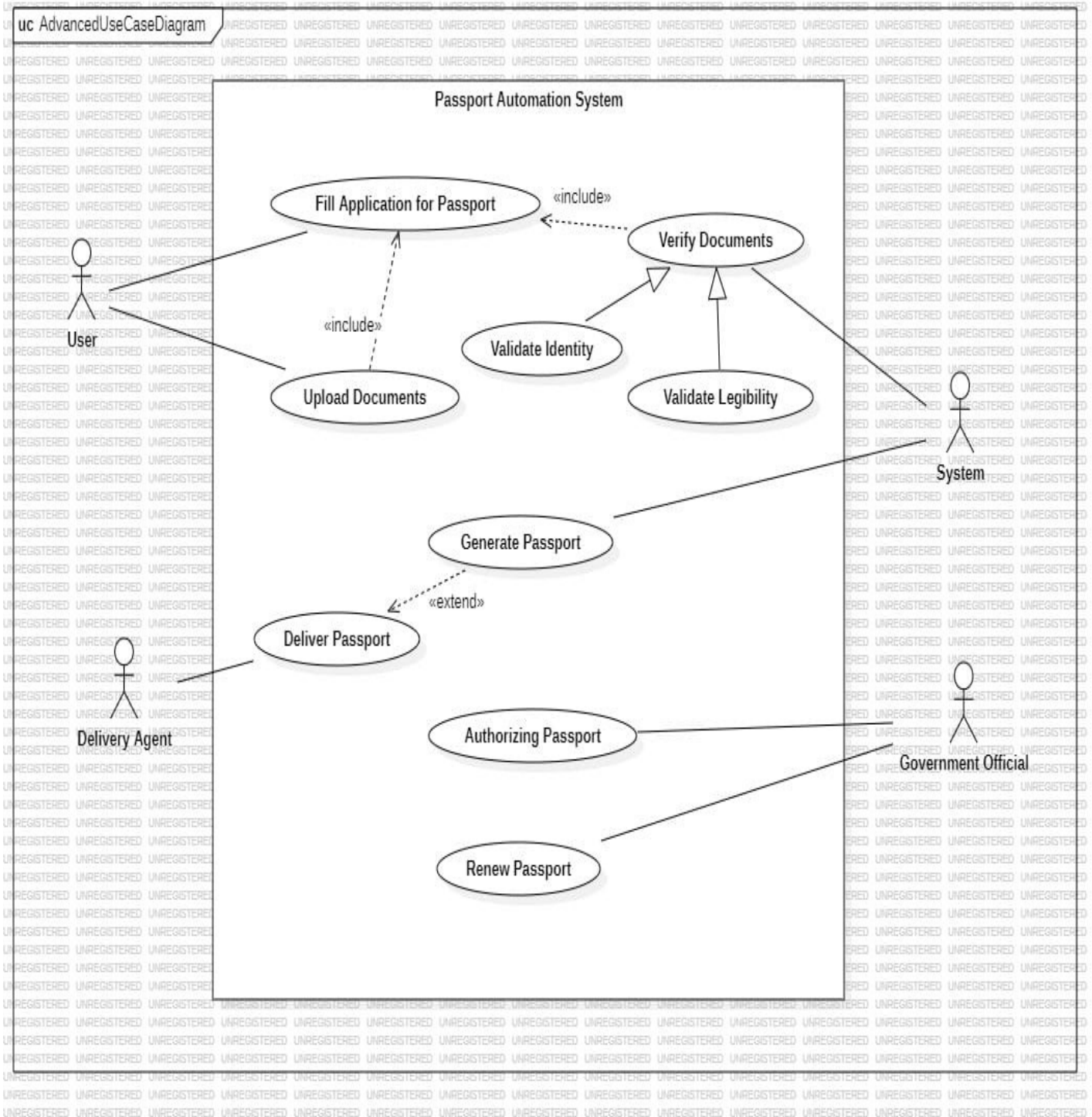
Advanced Class Diagram



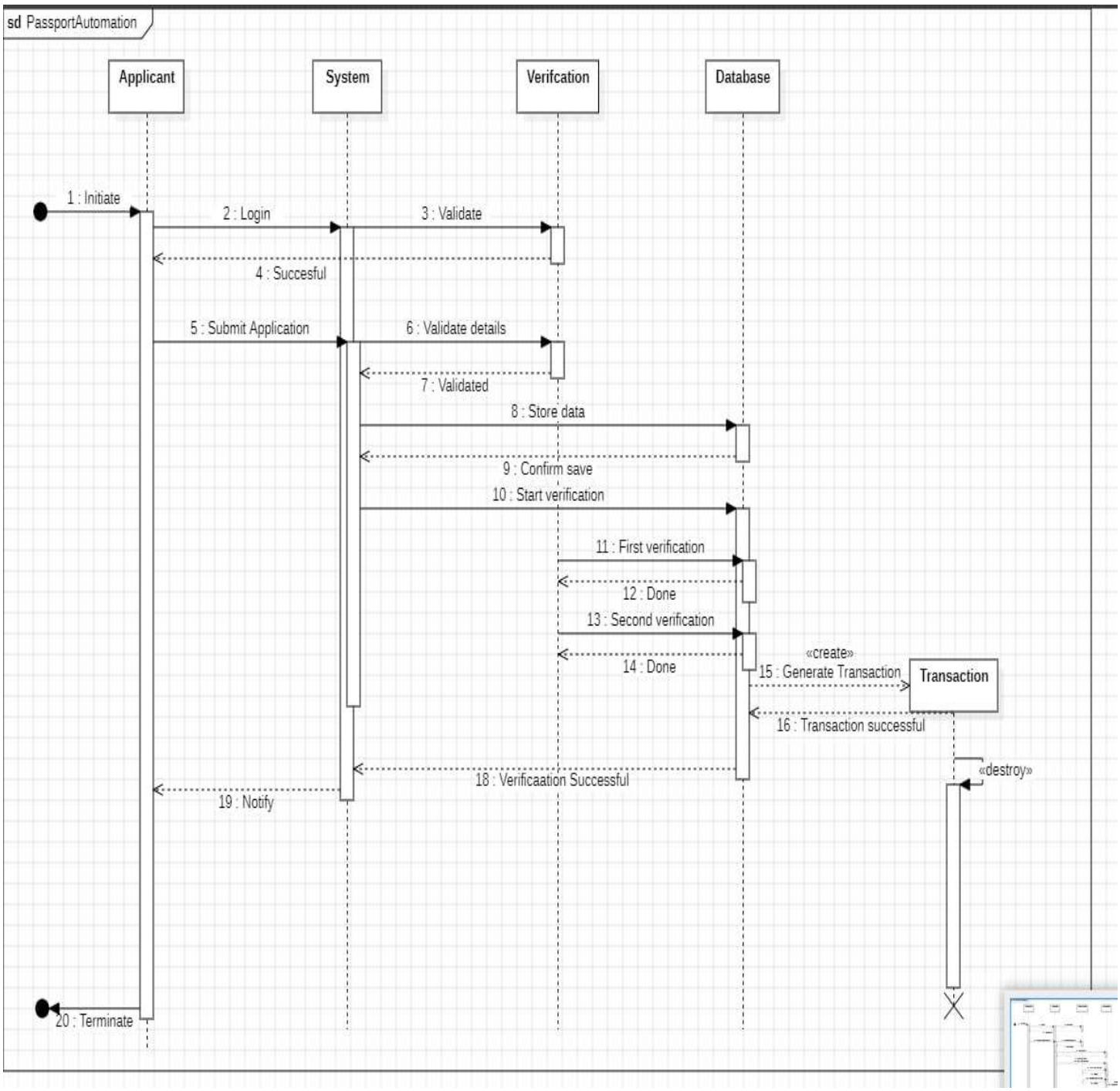
Advanced State Diagram



Advanced Use Case Diagram



Advanced Sequence Diagram



Advanced Activity Diagram

Activity1::PassportAutomationSystem