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:

SHASHANK C

1BM22CS254

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

Scanned with OKEN Scanner

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 **SHASHANK C(**1BM22CS254) during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

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

Scanned with OKEN Scanner

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

Scanned with OKEN Scanner

② Hotel Management System

→ Introduction.

• Purpose of the Document :- The primary purpose of this document is to outline the functional and non-functional requirements. This document will be used as foundation to guide developers during the design and design and implementation phases.

• Scope of the Document.
The system will manage various aspects of hotel operations, including room reservations, check-in/check-out processes, room availability tracking, billing, and customer information management.

• Overview :- The overview of the documents represents the basic key features of the Hotel Management system and also summarizes all the functionalities, system interfaces and assumptions and constraints.

→ General Description :- The description includes the streamline and automate various aspects of hotel operations. This system will enable hotel staff and management to efficiently manage room bookings, customer information, billing, housekeeping services, and other related functions.

→ Functional Requirements :-

* Room Reservation :- The system must allow users to search for available rooms, book them, modify bookings, and cancel reservations.
* check-in/check-out :- It should enable staff to check guests in and out, and update room availability in real-time.
* Billing and Payment :- The system must generate invoices and support multiple payment methods.
* Customer Data Management :- Maintaining a database of customer profiles, stay history, and preferences.

→ Interface Requirements :-

* User Interface :- The system must provide a web-based UI for hotel staff to manage bookings and services.
* Database interface :- It should interact with an RDBMS to store and retrieve the data.
* Payment Gateway :- Integration with external payment gateways for secure processing of online payments.

→ Performance Requirement :- The responsetime should be less as the system must respond to users actions, also the system must handle concurrent users that is, when 100 simultaneous users must experience an smoothness without any problem.

→ Design constraints :- It includes that the system must maintain data privacy as regulations such as GDPR ensuring secure handling. The system must use a RDBMS for storing data and the system must be compatible with windows, macOs and Linux for desktop users, and standard web browsers for web access.

→ Non-Functional Requirements.

* Performance :- The system must handle multiple concurrent users without degradation in performance.
* Reliability :- It should have an uptime of 99.9%, ensuring availability for critical operations.
* Security :- Must implement encryption for sensitive data and role-based access control.
* Scalability :- The system should support scalability to accomodate growing hotel chains or increased room capacity.

→ Preliminary schedule and Budget :-
The estimated timeline is 6 months, and is divided as requirement analysis (2 weeks), system design (4 weeks), development (12 weeks), Testing (6 weeks), deployment and training (4 weeks) and remaining for Post-Deployment support that is 2 weeks.
The Budget is approximately given as $100,000, where the software developement is $60,000, Hardware and Infrastructure $15,000, Licensing and software Tools is $10,000, Training and Maintainance includes $15,000.

Scanned with OKEN Scanner

Class Diagram

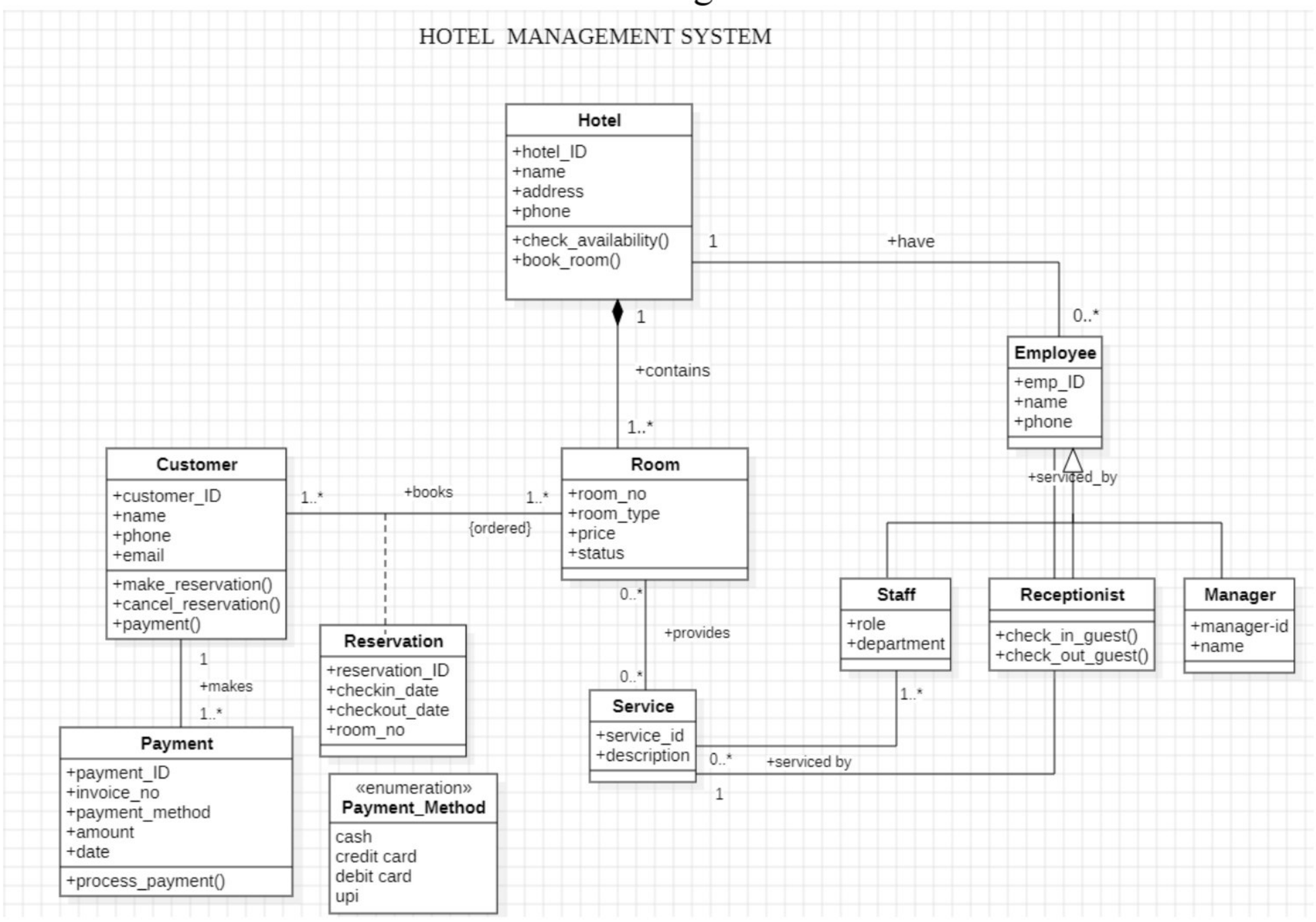


Fig1.1 Hotel Management System - Class Diagram

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

Scanned with OKEN Scanner

State Diagram

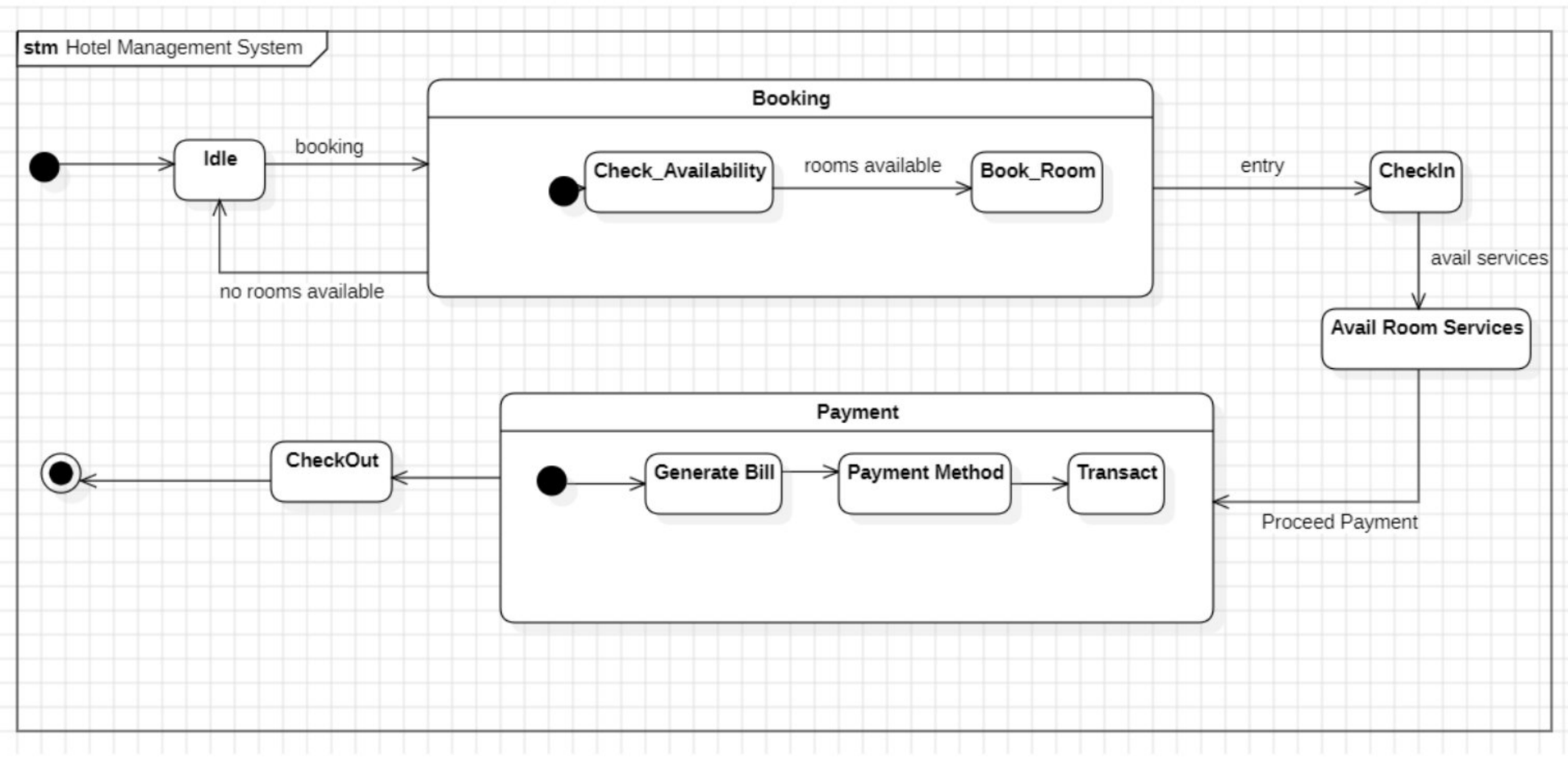


Fig1.2 Hotel Management System - State Diagram

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.

Scanned with OKEN Scanner

Use Case Diagram

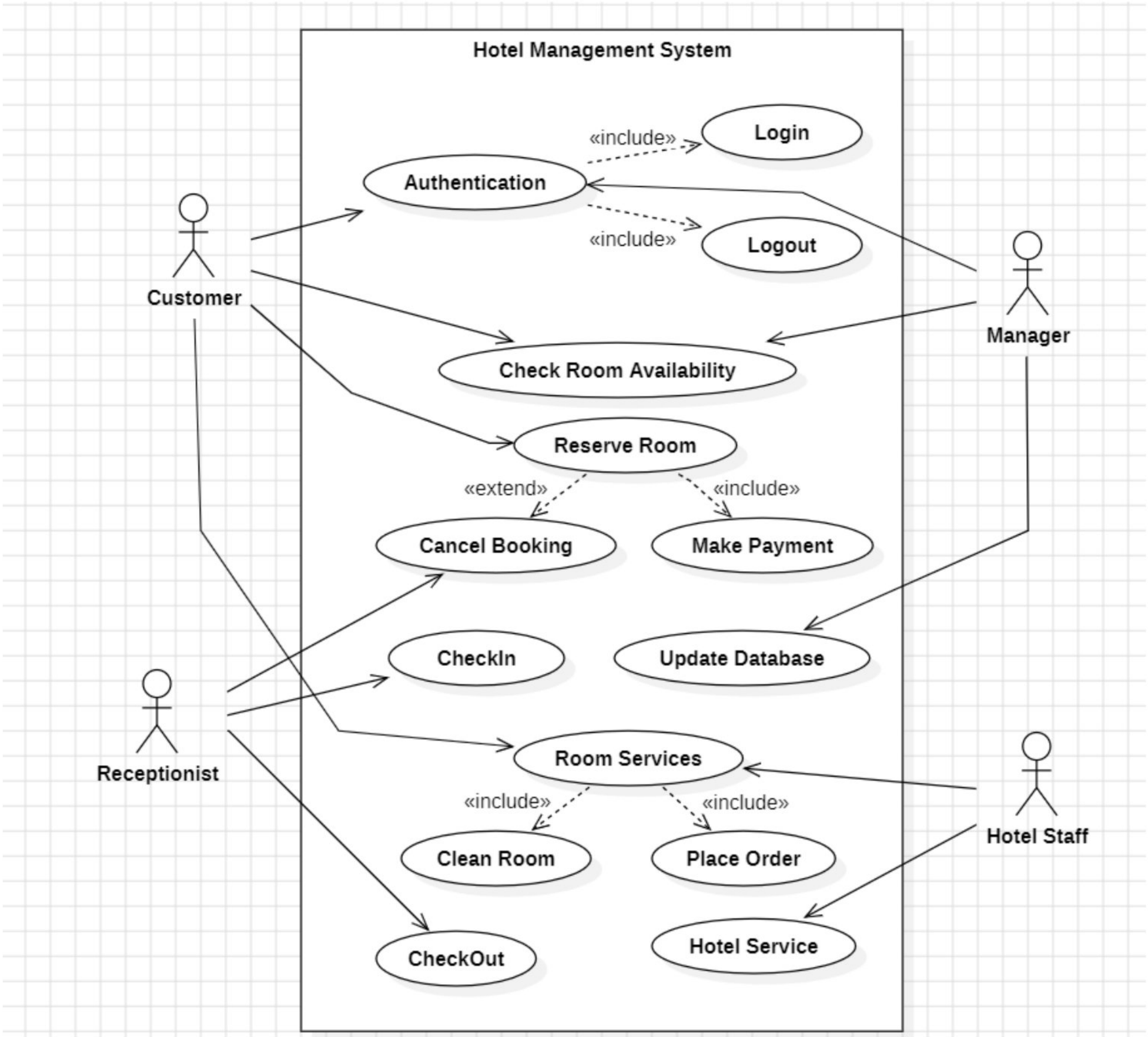


Fig1.3 Hotel Management System - Use Case Diagram

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

Scanned with OKEN Scanner

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

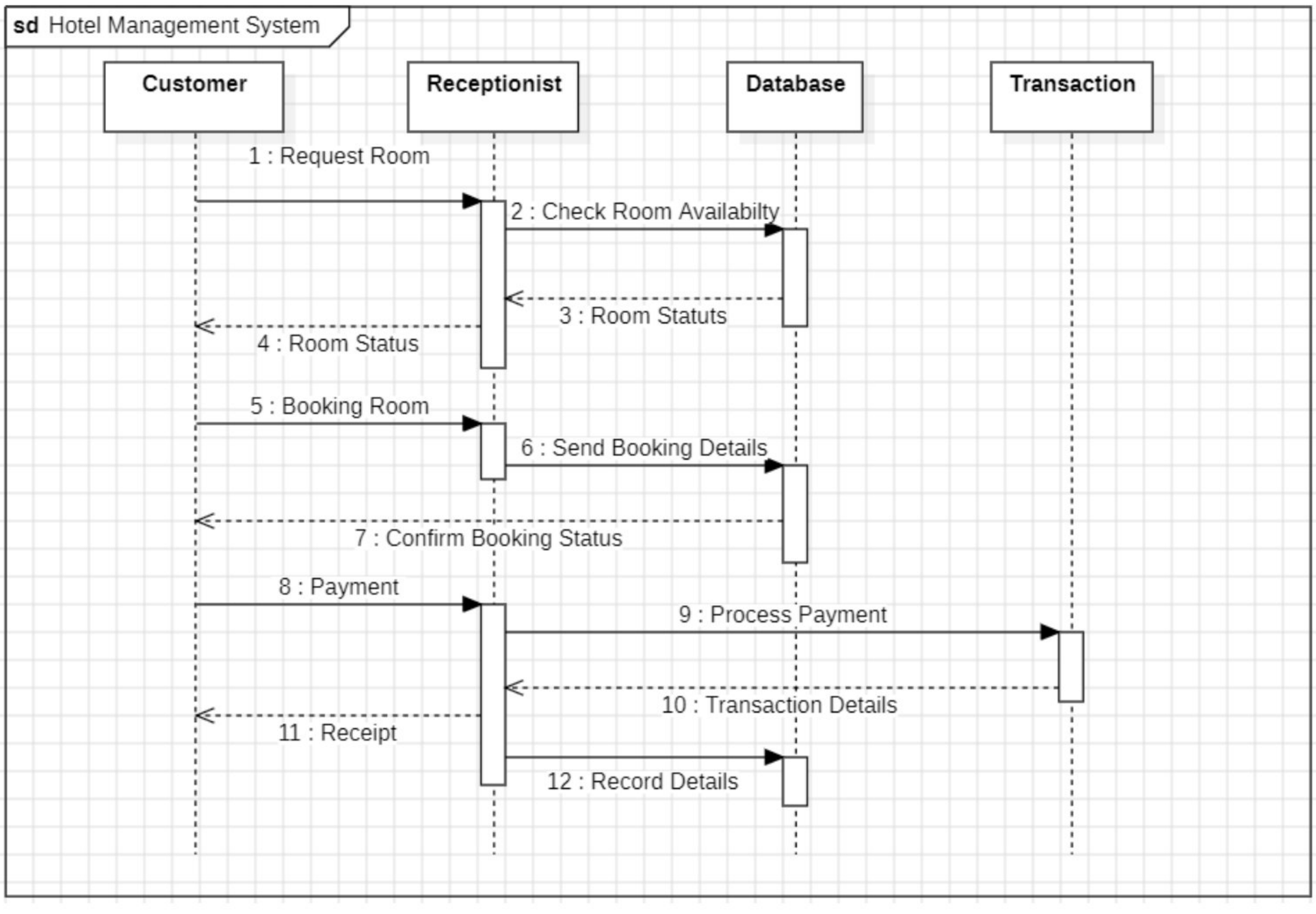


Fig1.4 Hotel Management System - Sequence Diagram

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

Scanned with OKEN Scanner

the database. Finally, the customer receives a receipt, and the database records the transaction details.

Activity Diagram

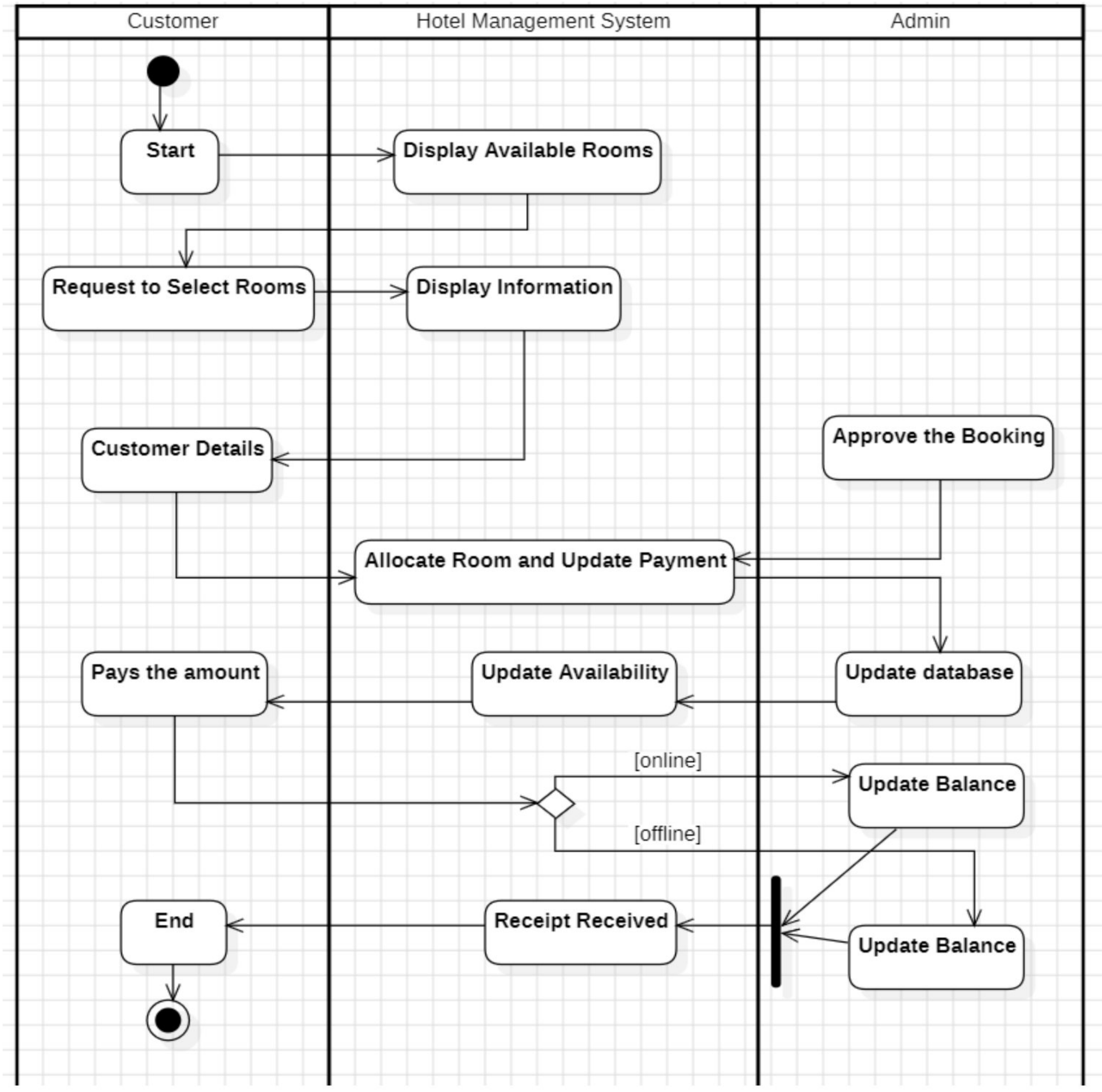


Fig1.5 Hotel Management System - Activity Diagram

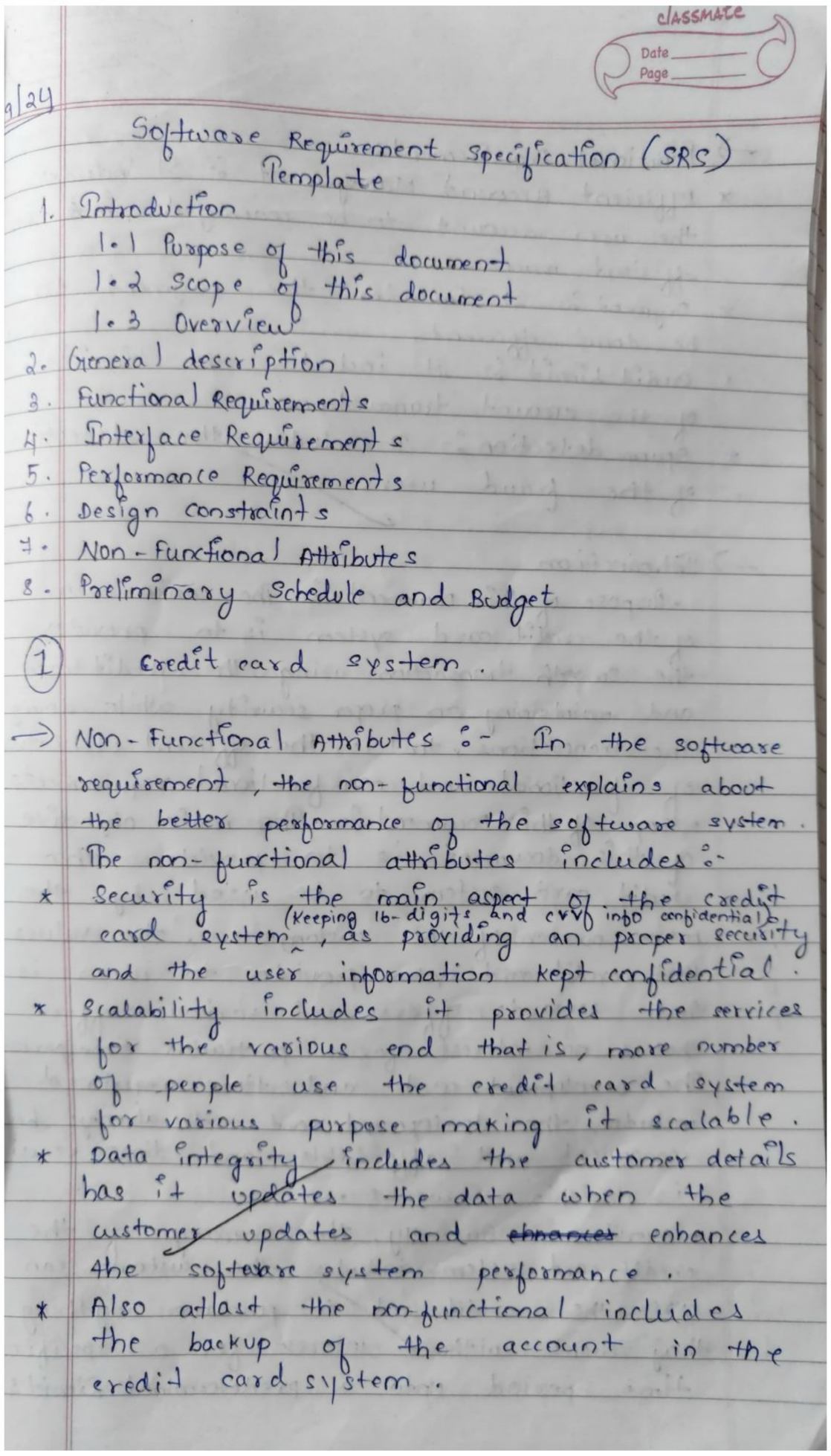
The activity diagram illustrates the process of booking a room at a hotel. The customer starts by requesting to select rooms, and the system displays available rooms and their information. The

Scanned with OKEN Scanner

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



9/24

Software Requirement specification (SRS) Template

1. Introduction
   - 1.1 Purpose of this document
   - 1.2 Scope of this document
   - 1.3 Overview
2. General description
3. Functional Requirements
4. Interface Requirements
5. Performance Requirements
6. Design constraints
7. Non-Functional Attributes
8. Preliminary Schedule and Budget

(1) Credit card system.

→ Non-Functional Attributes :- In the software requirement, the non-functional explains about the better performance of the software system. The non-functional attributes includes :-

* Security is the main aspect of the credit card system, as providing an proper security (Keeping 16-digits and cvv info confidential) and the user information kept confidential.
* Scalability includes it provides the services for the various end that is, more number of people use the credit card system for various purpose making it scalable.
* Data integrity includes the customer details has it updates the data when the customer updates and ~~ehnances~~ enhances the software system performance.
* Also atlast the non-functional includes the backup of the account in the credit card system.

→ Functional Requirements :-

* Efficient Account Management :- It includes the user account to be managed in an efficient manner.
* Payment :- It includes the payment to be done efficient
* Credit Limit :- It includes the limits of the amount transactions.
* Spam detection :- It includes the detection of the fraud users.

→ Introduction

- Purpose of this Doument :- The main aim of the credit card system is to provides the smooth transactions using the credit card and maintaining an proper security while doing the transactions, that is, the efficient utilization of functional and non-functional requirements.
- Scope of this Doument :- The main objective of this document is to provide the credit card system to be used by the more number of the customers, the values that credit card system provides to users is that rewards, offers and various offers that is no interest for a partiular time period. The development cost and time required includes, the development cost will always be economical and affordable and time required is within 2 weeks.
- overview :- Basically, the summary of the credit card system includes the customer can use the credit card to purchase anything they want without interest for a specific time period upto a predetermined limits.

→ General Description :- The description includes the objective of explaining the user features which are maintaining account balance and reviewing about all the transaction made (transaction history) and also includes the features of the user community.

→ Interface Requirements :- The interface requirements includes the authorised bank websites with communicates with the users by providing account details, mobile number and other details to get access to the authorised website.

→ Performance Requirements :- It includes the website should handle the load that is, more users using the website should use with ease, and more detailed explanation includes the app must be maintained properly handling various users.

→ Design constraint :- It includes the rules of the RBI must be followed, the interface must be similar of the UPI and rules specified in an authorized document should be followed.

→ Preliminary schedule and Budget.
The initial budget is expected to be $100000 and expected date is to end on 1st week of February. The initial version just includes the features and there is no rewards and credit score.

30/9/2024

Class Diagram

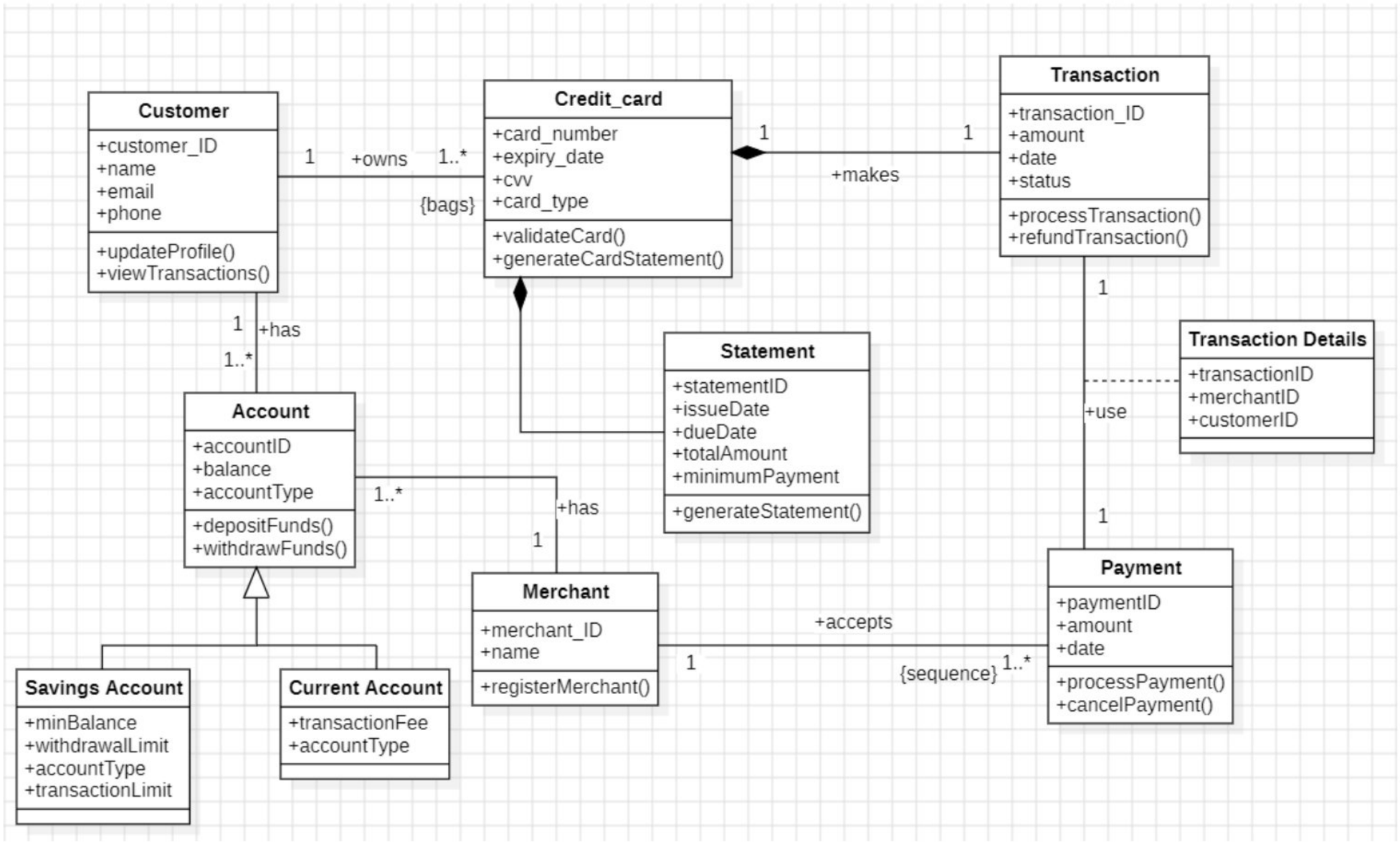


Fig 2.1 Credit Card Processing System - Class Diagram

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.

Scanned with OKEN Scanner

State Diagram

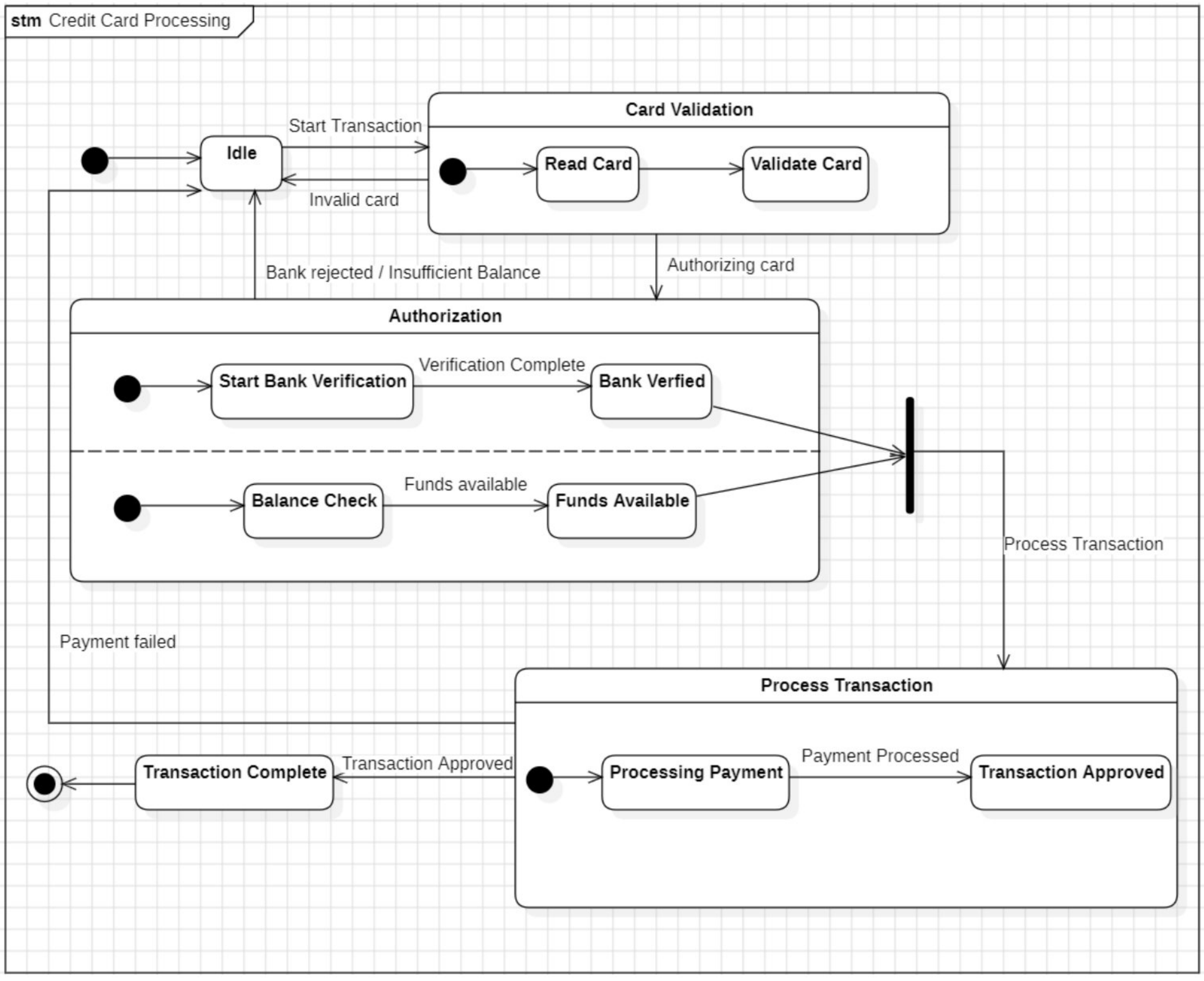


Fig 2.1 Credit Card Processing System - State Diagram

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.

Scanned with OKEN Scanner

Use Case Diagram

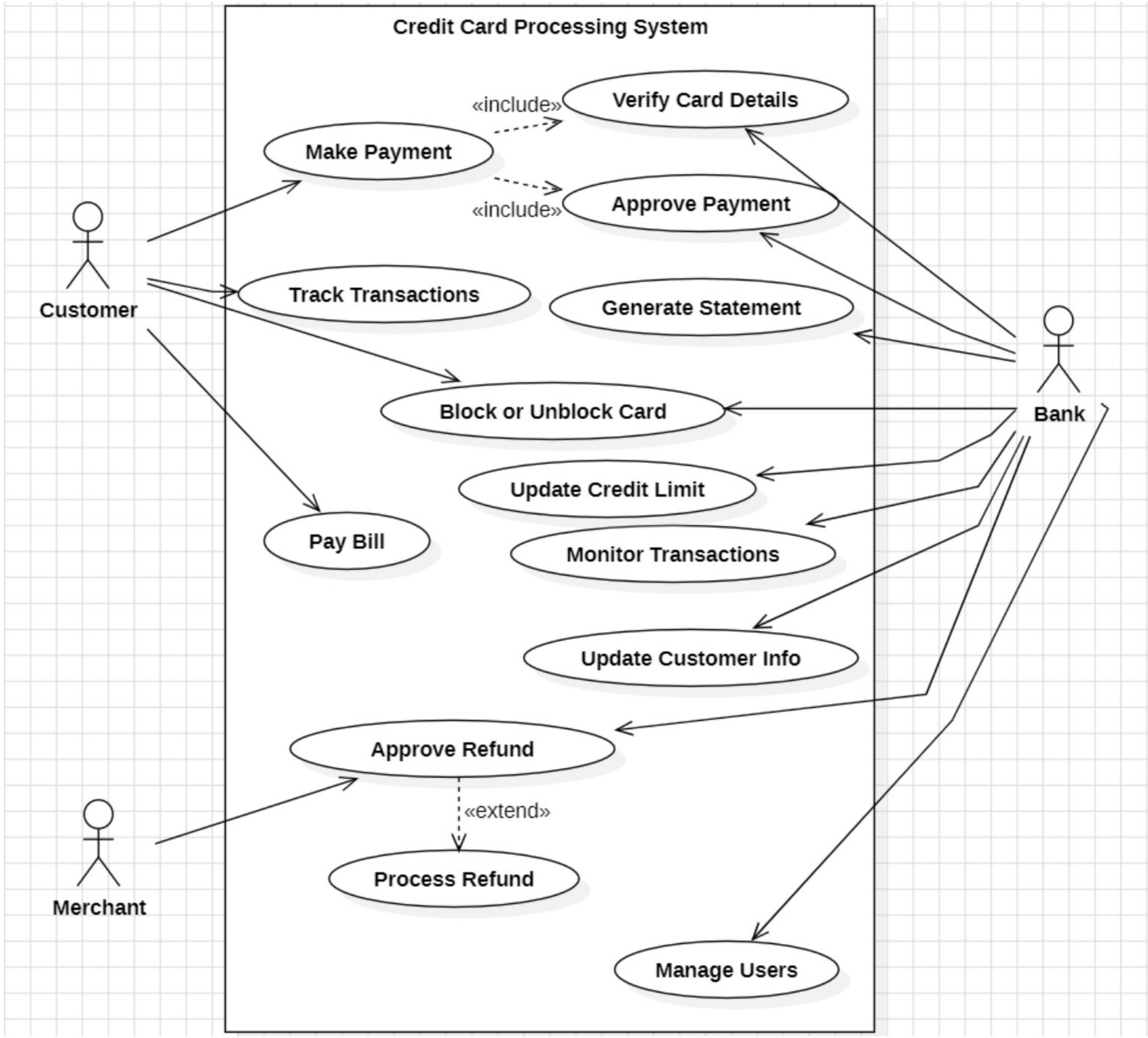


Fig 2.1 Credit Card Processing System - Use Case Diagram

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.

Scanned with OKEN Scanner

Sequence Diagram

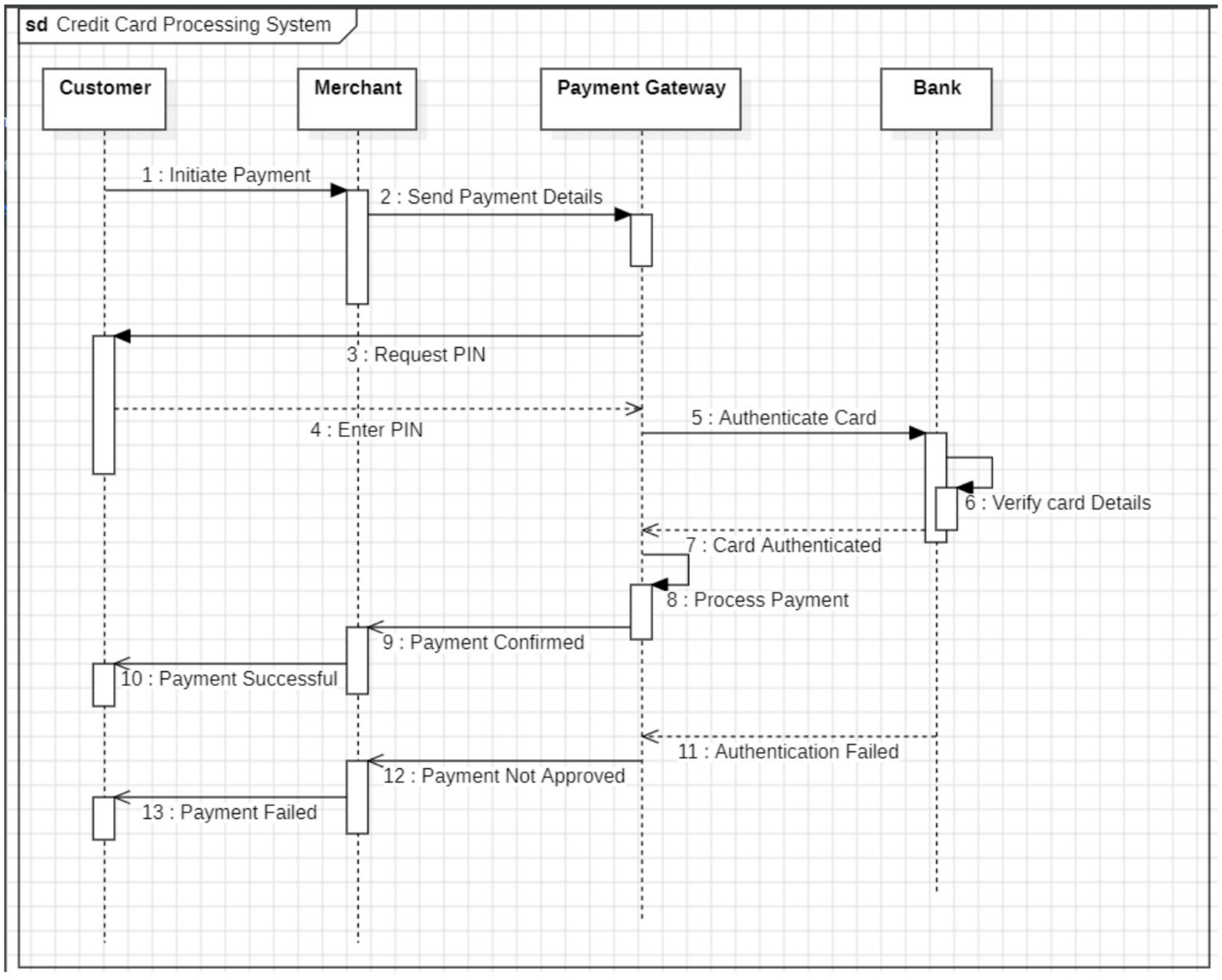


Fig 2.1 Credit Card Processing System - Sequence Diagram

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

Scanned with OKEN Scanner

Activity Diagram

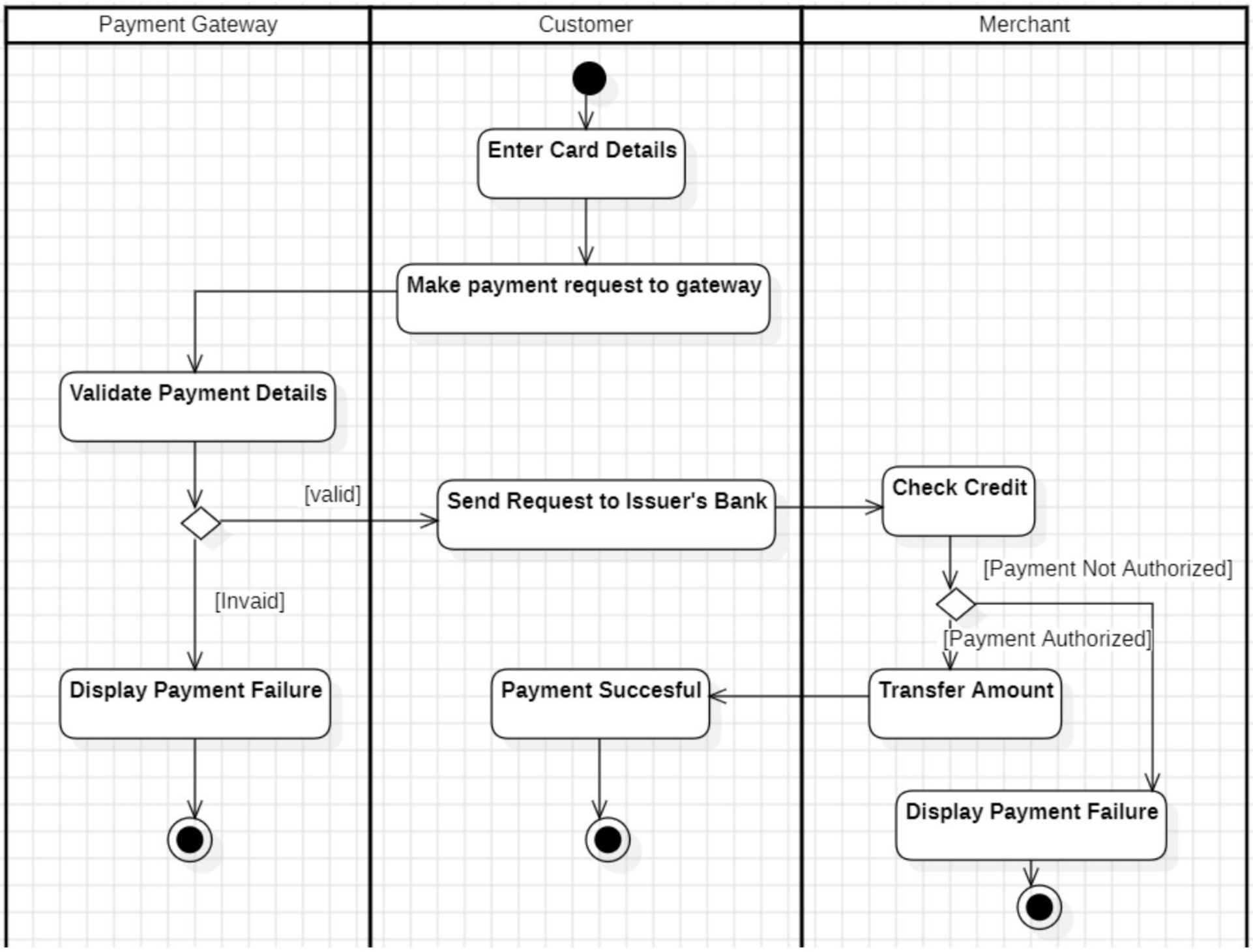


Fig 2.5 Credit Card Processing System - Activity Diagram

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.

Scanned with OKEN Scanner

3. Library Management System

Software Requirement Specification

7/10/24

(3) Library Management system

→ Introduction

- Purpose of the document :- This document is to define the requirements for developing, and it details the system's functionality, performance, and design specifications to guide the development team.
- Scope of the Document :- This document outlines the features, functionalities and technical requirements. It covers aspect like book management, member registrations, borrowing and returning processes.
- Overview :- The overview includes the main features, user classes, and technical architecture. It is organized into sections detailing functional and non-functional requirements.

→ General Description :- The description includes library operations such as book borrowing, returns, cataloging, and fines. Users include librarians, members, and administrators, it will provide real-time update on book availability and track borrowing history.

→ Functional Requirements :-

* Book catalog Management :- The system must allow librarians to add, update, and remove books.
* Member Management :- It should handle member registration, profile updates, and member categorization.
* Borrow/Return :- The system must track book loans & returns, automatically updating availability.
* Fine calculation :- It should calculate fines for overdue books based on predefined rules.

→ Interface Requirements :-

* User Interface :- web-based interfaces for users and librarians to manage the library.
* Data interface :- The system will interface with a relational database to store book and user information.
* Barcode Reader :- Integrate with barcode readers for quick book checkouts & returns.

→ Performance Requirements :- The system must handle multiple users simultaneously without any degrade in performance, also data processing for transactions should process in under 1 second, also the real-time updates for book status and availability should occur quickly.

→ Non-Functional Requirements :-

* Performance :- The system must respond to user actions within 3 seconds.
* Scalability :- Must support upto 200 concurrent users.
* Security :- User & book data must be encrypted.
* Reliability :- The system should maintain 99.9% uptime for critical operations.

→ Design Constraints :- The systems must work on window and Linux for desktop use, and also a database to maintain the data, must adhere to data protection regulations like GDPR for member data. Barcode scanner support should be compatible with standard barcode scanning devices.

Scanned with OKEN Scanner

→ Preliminary schedule and Budget :- The schedule includes: Requirement Analysis (2 weeks) Design (3 weeks), Development (10 weeks) and Testing (4 weeks).
The Budget can be divided as development cost ($50,000), Hardware ($10,000), Soft Licenses ($5,000), Training and Maintainance for $10,000.

Scanned with OKEN Scanner

Class Diagram

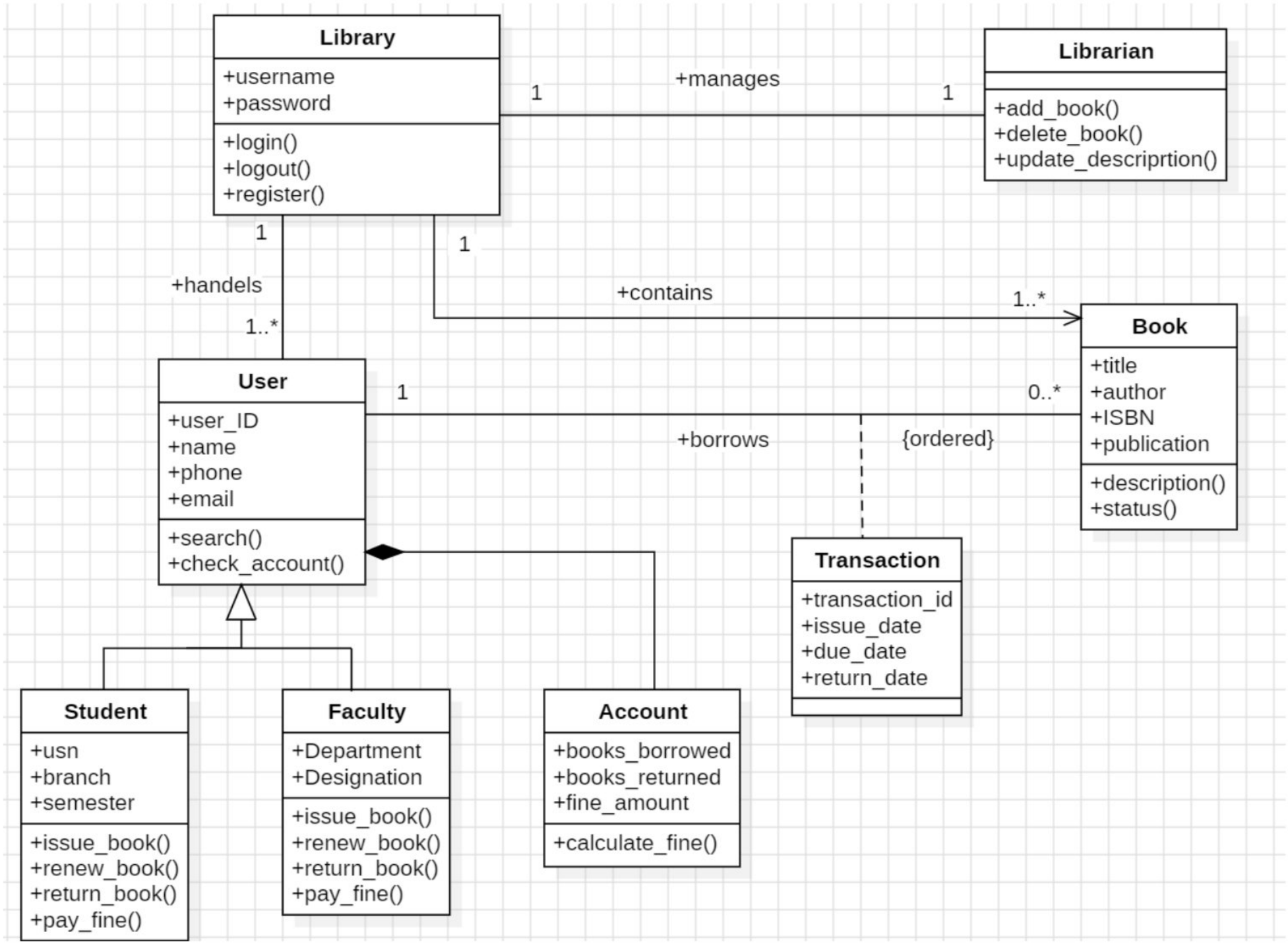


Fig 3.1 Library Management System -Class Diagram

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.

Scanned with OKEN Scanner

State Diagram

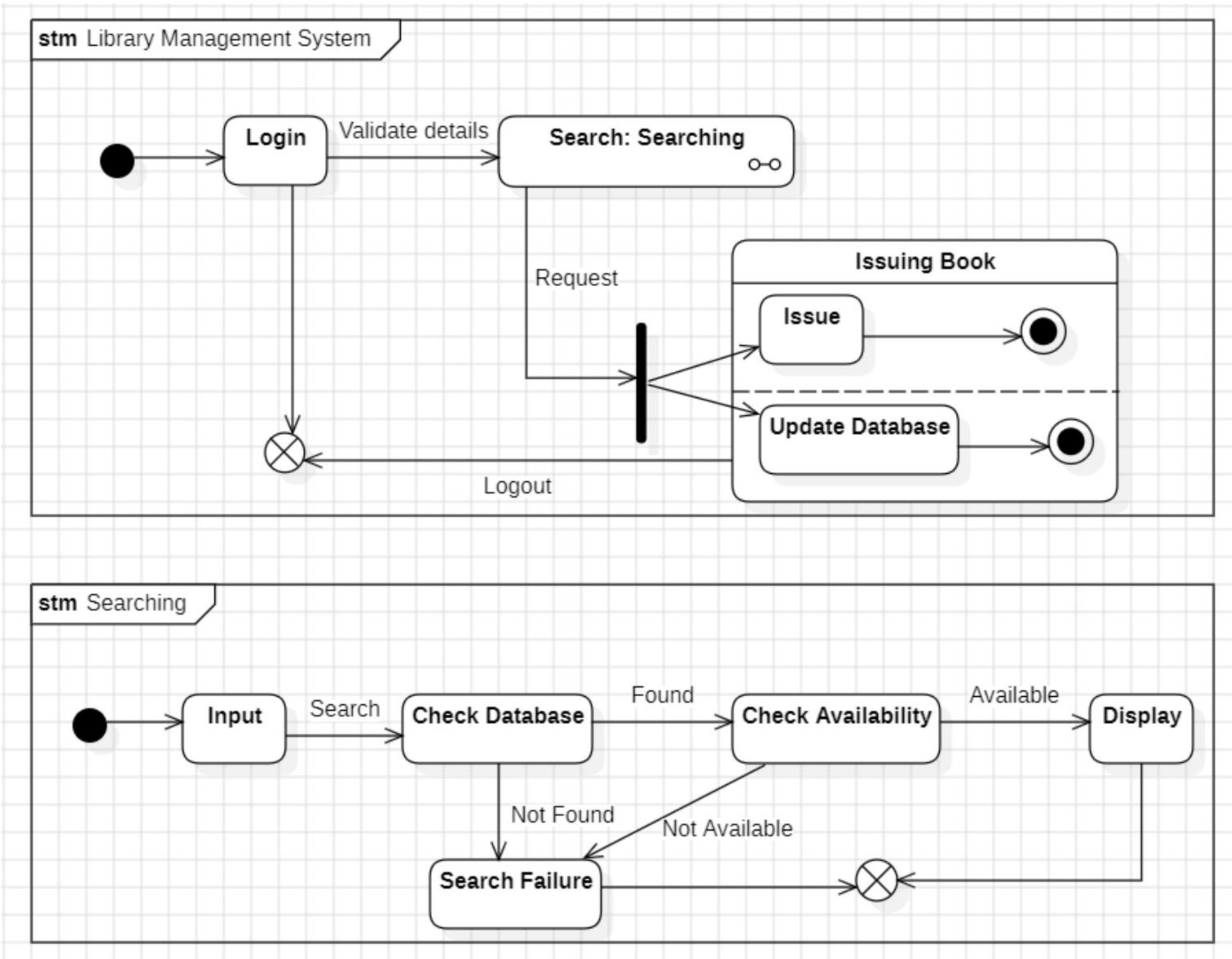


Fig 3.2 Library Management System - State Diagram

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.

Scanned with OKEN Scanner

Use Case Diagram

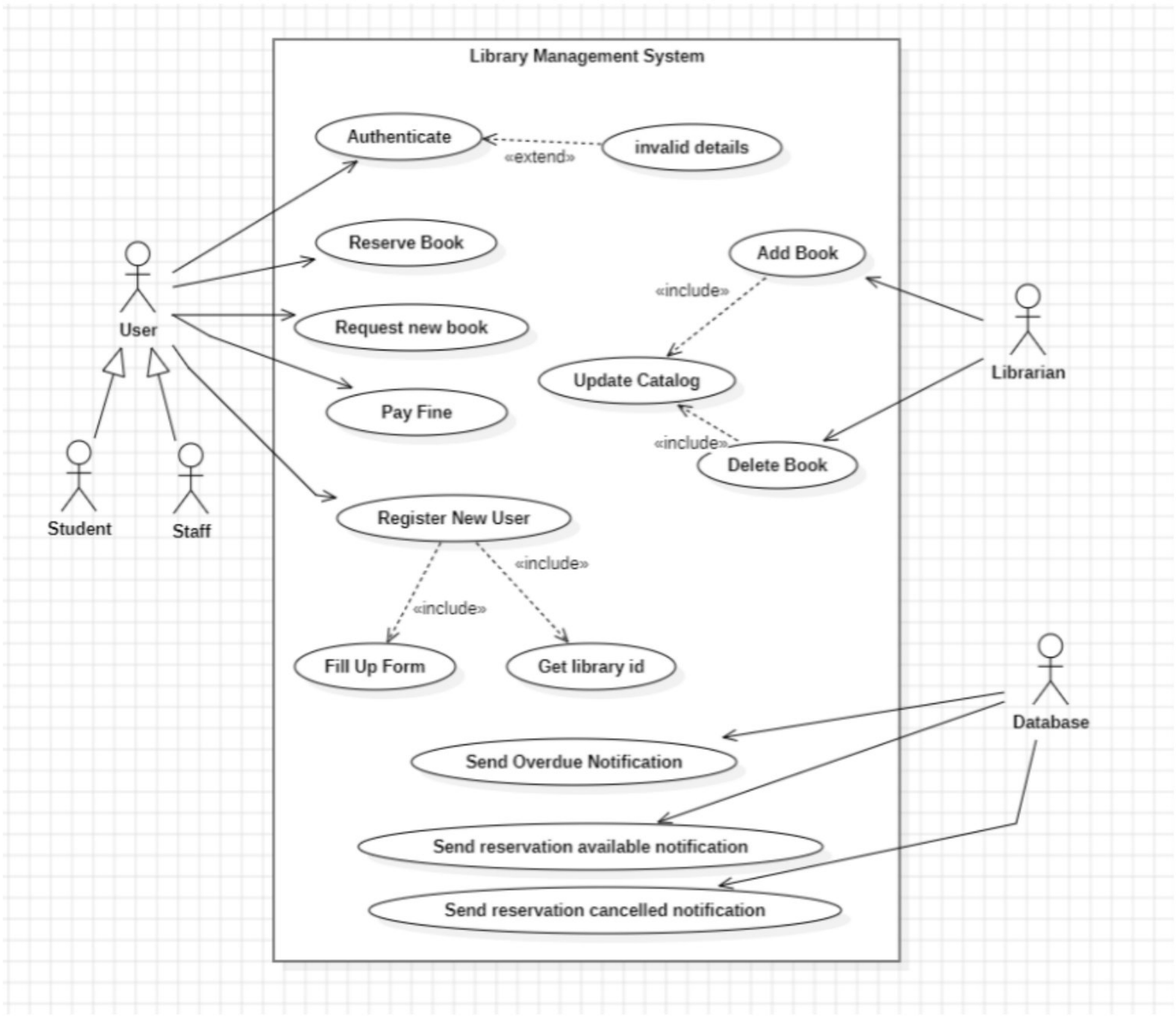


Fig 3.3 Library Management System - Use Case Diagram

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.

Scanned with OKEN Scanner

Sequence Diagram

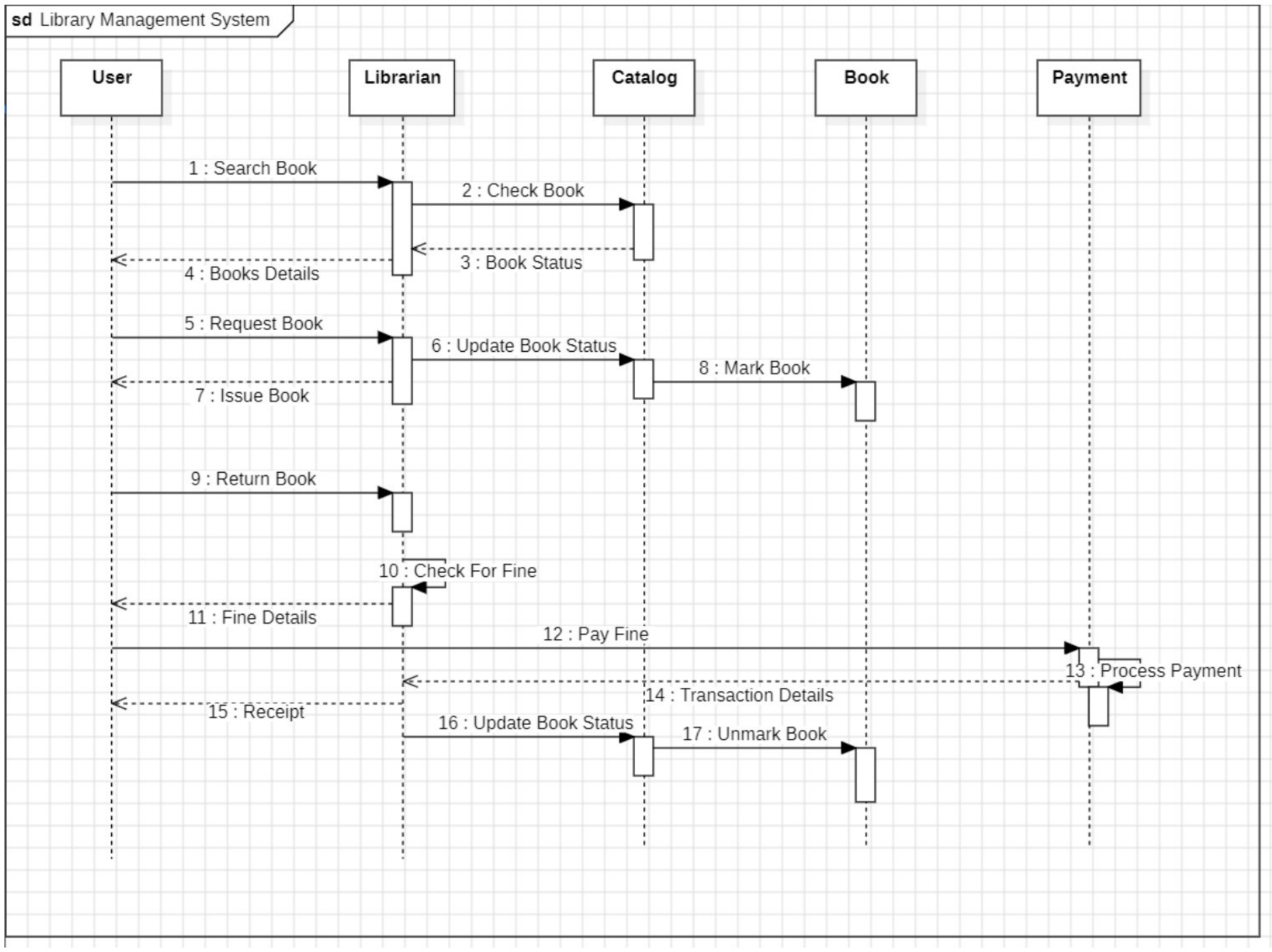


Fig 3.4 Library Management System - Sequence Diagram

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.

Scanned with OKEN Scanner

Activity Diagram

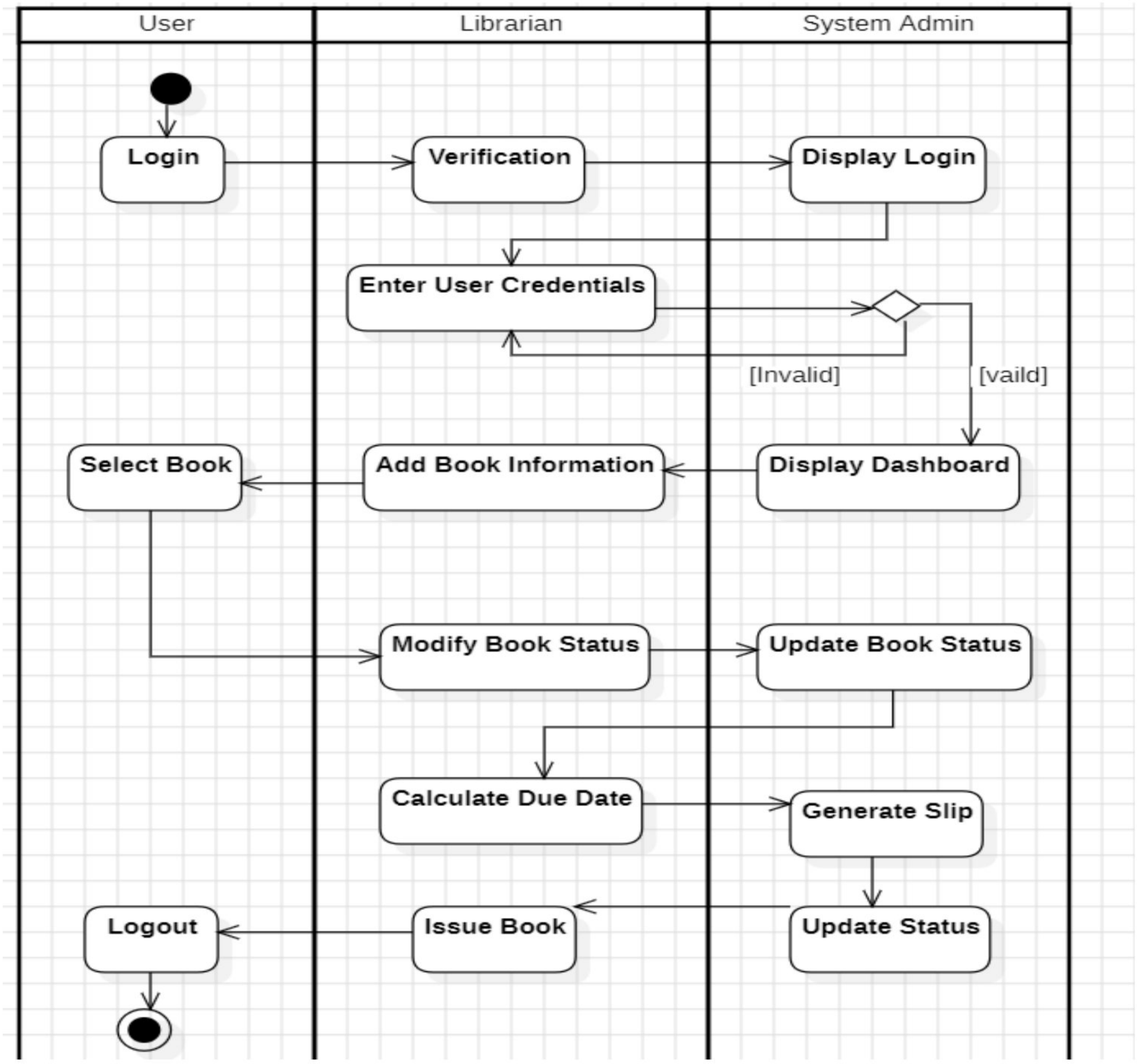


Fig 3.5 Library Management System - Activity Diagram

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.

Scanned with OKEN Scanner

4. Stock Maintenance System

Software Requirement Specification

4 Stock Maintenance System

→ Introduction

- Purpose of the Document :- The document is to provide detailed guidance for designing a system that helps manage inventory, stock levels, purchase, and sales.
- Scope of the Document :- It covers essential features such as stock level management, order processing, and inventory reporting. It also defines user roles and system interfaces but excludes third-party integrations.
- Overview :- The document gives primary functions, including inventory tracking, sales management, and purchase monitoring. It also defines functional, non-functional, interfaces and performance requirements.

→ General Description :- The system is designed to manage inventory effectively by automating key processes such as stock tracking, order management, and inventory reporting. The system will allow users to monitor stock levels in real-time, generate purchase orders, and analyze sales data.

Scanned with OKEN Scanner

→ Functional Requirements :-

* Inventory Management :- The system must allow users to add, update, and delete stock items, including detailed such as quantity & price.
* stock Tracking :- It should monitor stock levels in real-time, alerting users when items fall below predefined thresholds.
* Order Management :- The system must enable users to create, modify, and track purchase orders and sales orders.

→ Interface Requirements :-

* User-interface :- The system must provide a web-based interface for users to manage stock and inventory easily.
* Database interface :- It should interface with RDBMS to store & retrieve data.
* Barcode Scanner support :- Integration with barcode scanning devices for quick stock entry and inventory checks.

→ Performance Requirements :- The system must handle multiple users concurrently without any degrade in the performance, and the complete actions like searching and updating stock must be done quickly.

→ Design constraints :- The system must operate on Windows and Linux platforms. The system must comply with data proteetion, ensuring secure handling of user and stock data. It should be compatible with standard barcode scanners and RFID systems for inventory tracking.

→ Non-Functional Requirements :-

* Performance :- The system should respond to user actions within 2 seconds and handle multiple users.
* security :- The system must implement role-based access control and encrypt sensitive data.
* scalability :- It should be designed to accomodate future growth, allowing for additional users and increased stock items without impacting performance.

→ Preliminary schedule and Budget :- The schedule includes the Requirement Analysis (2weeks), Design (4 weeks), Development (10weeks), Testing and Deployment Training (6 weeks). The Budget is divided into Development costs $40,000, Hardware is $8,000, software Licenses is $5000, Training and Maintenance is $8,000.

Scanned with OKEN Scanner

Class Diagram

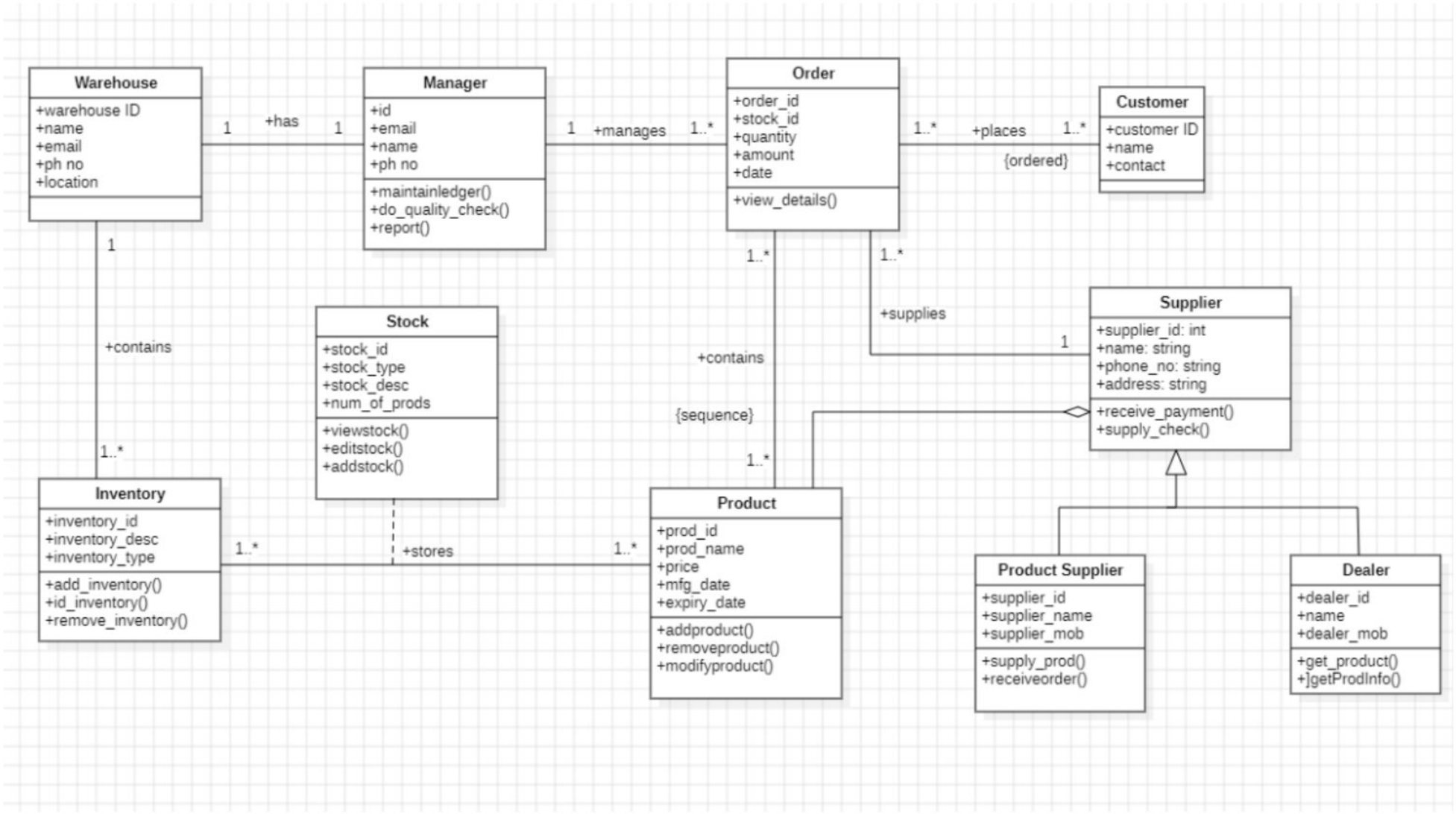


Fig 4.1 Stock Maintenance System - Class Diagram

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.

Scanned with OKEN Scanner

State Diagram

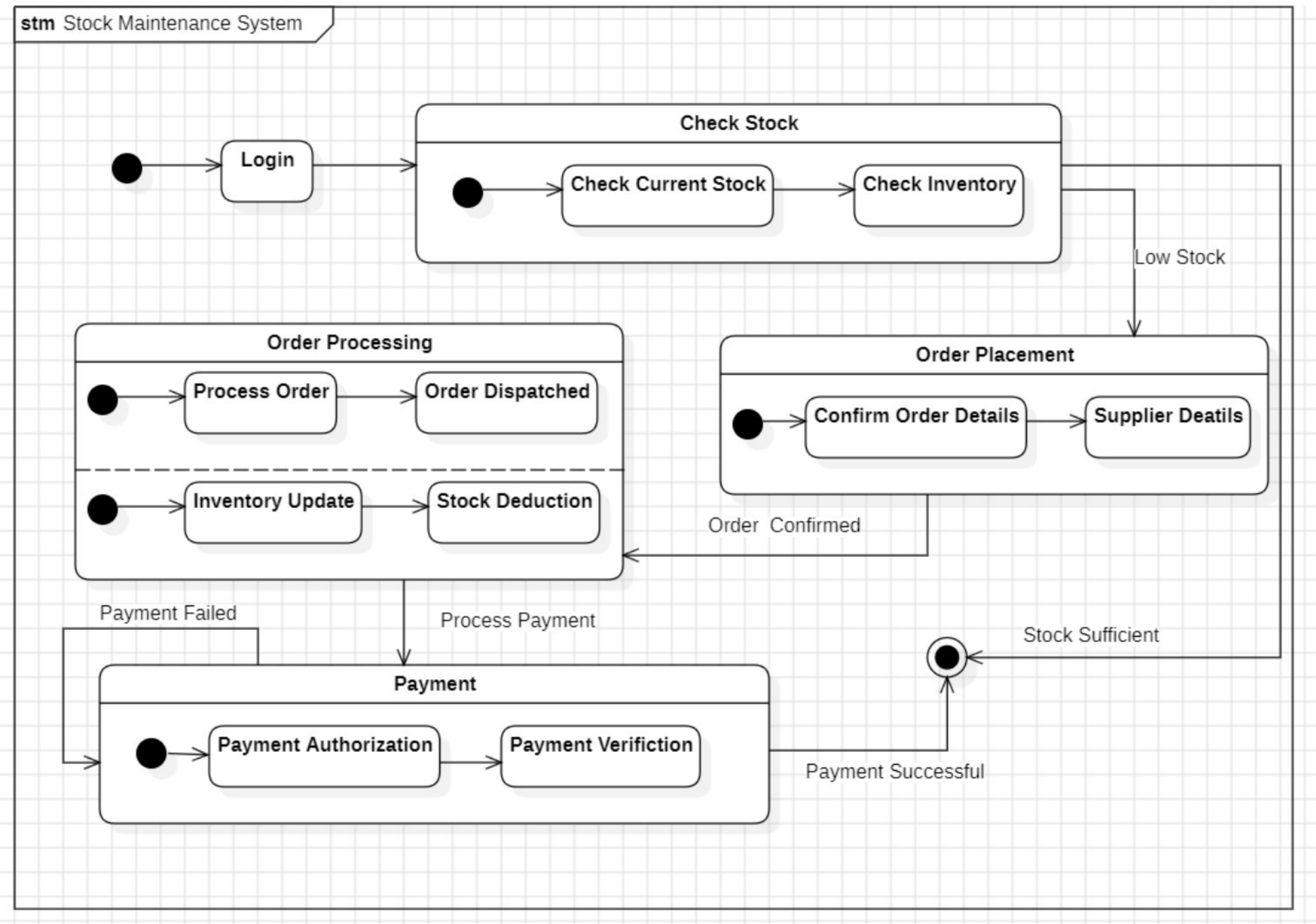


Fig 4.2 Stock Maintenance System - State Diagram

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.

Scanned with OKEN Scanner

Use Case Diagram

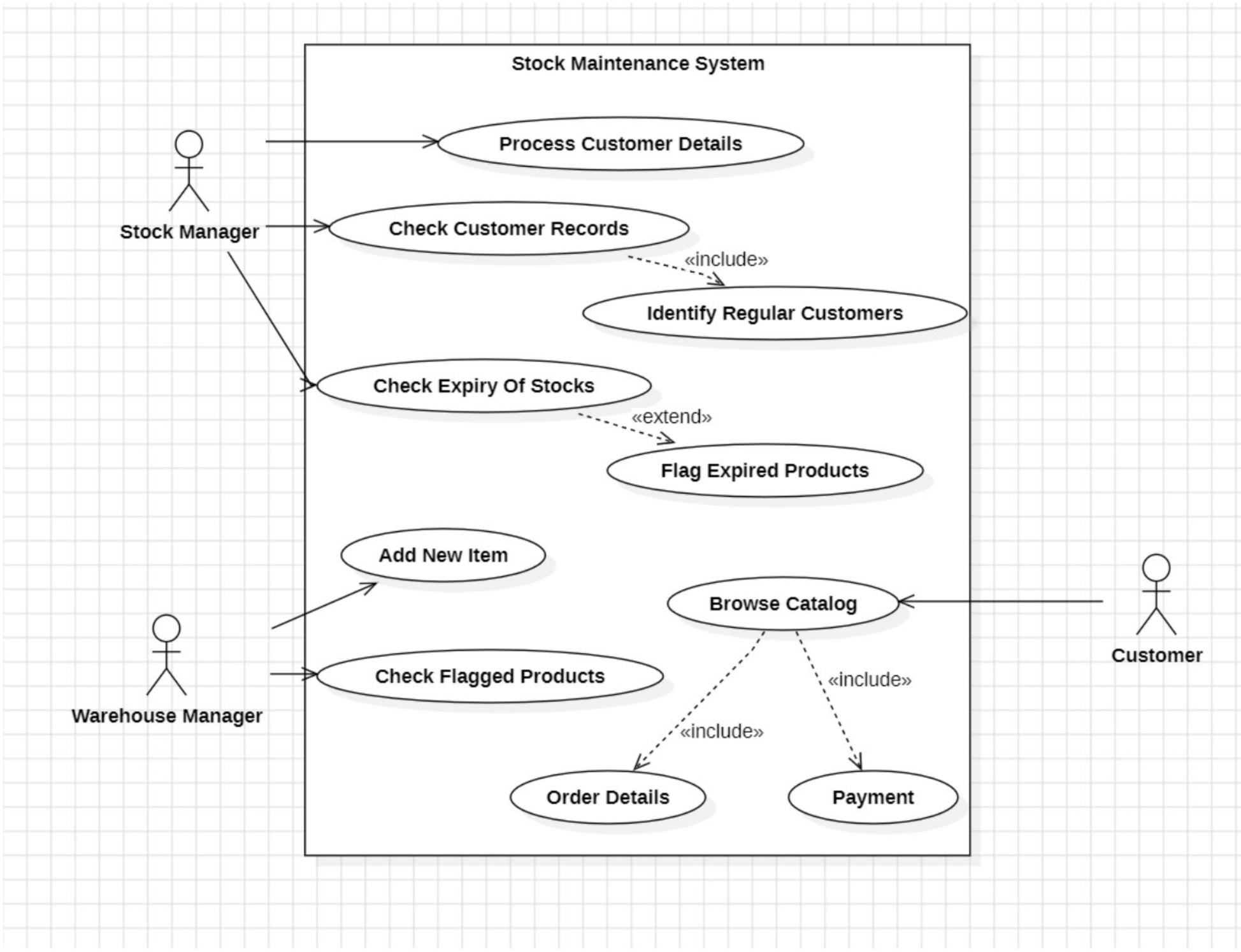


Fig 4.3 Stock Maintenance System - Use Case Diagram

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.

Scanned with OKEN Scanner

Sequence Diagram

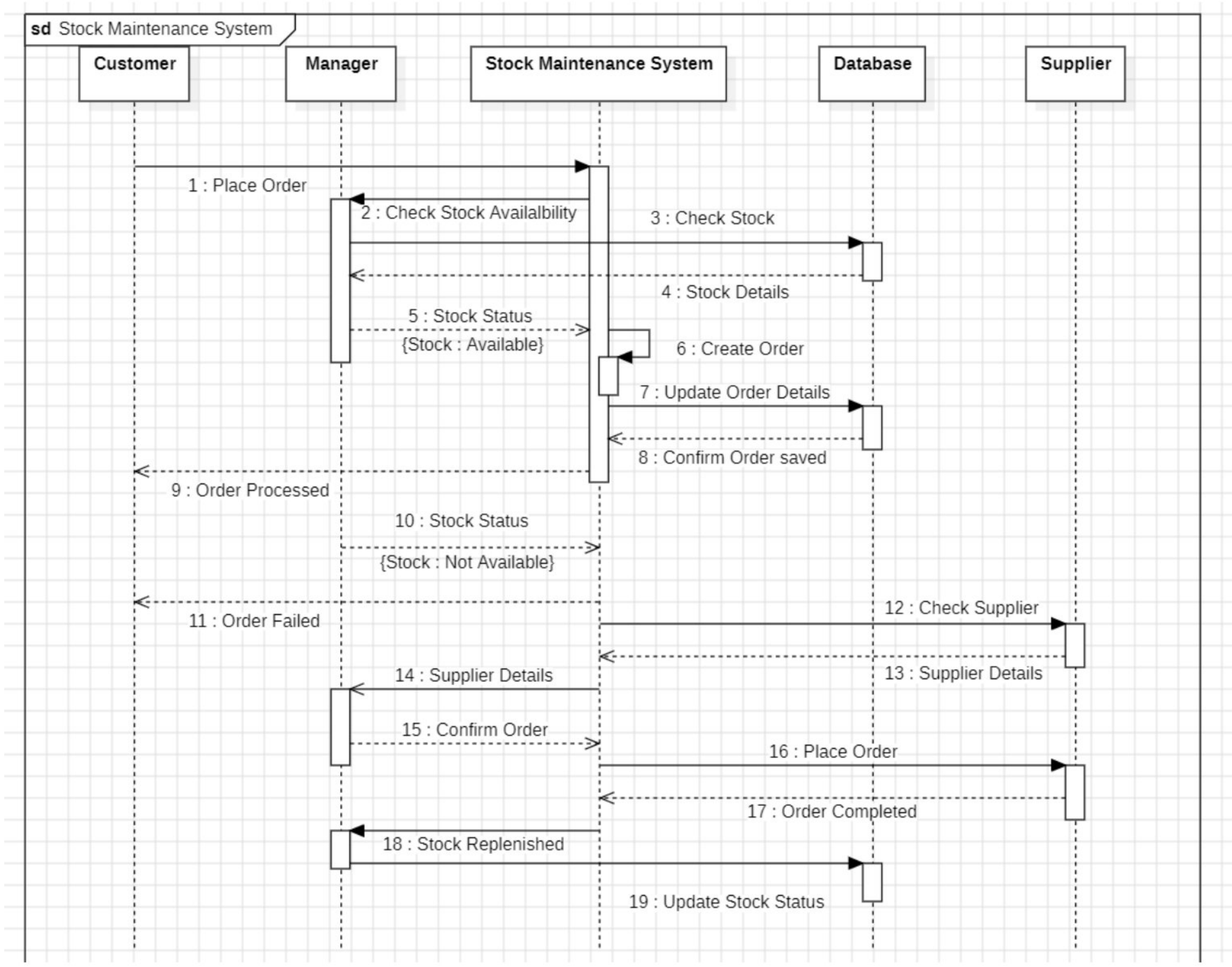


Fig 4.4 Stock Maintenance System - Sequence Diagram

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.

Scanned with OKEN Scanner

Activity Diagram

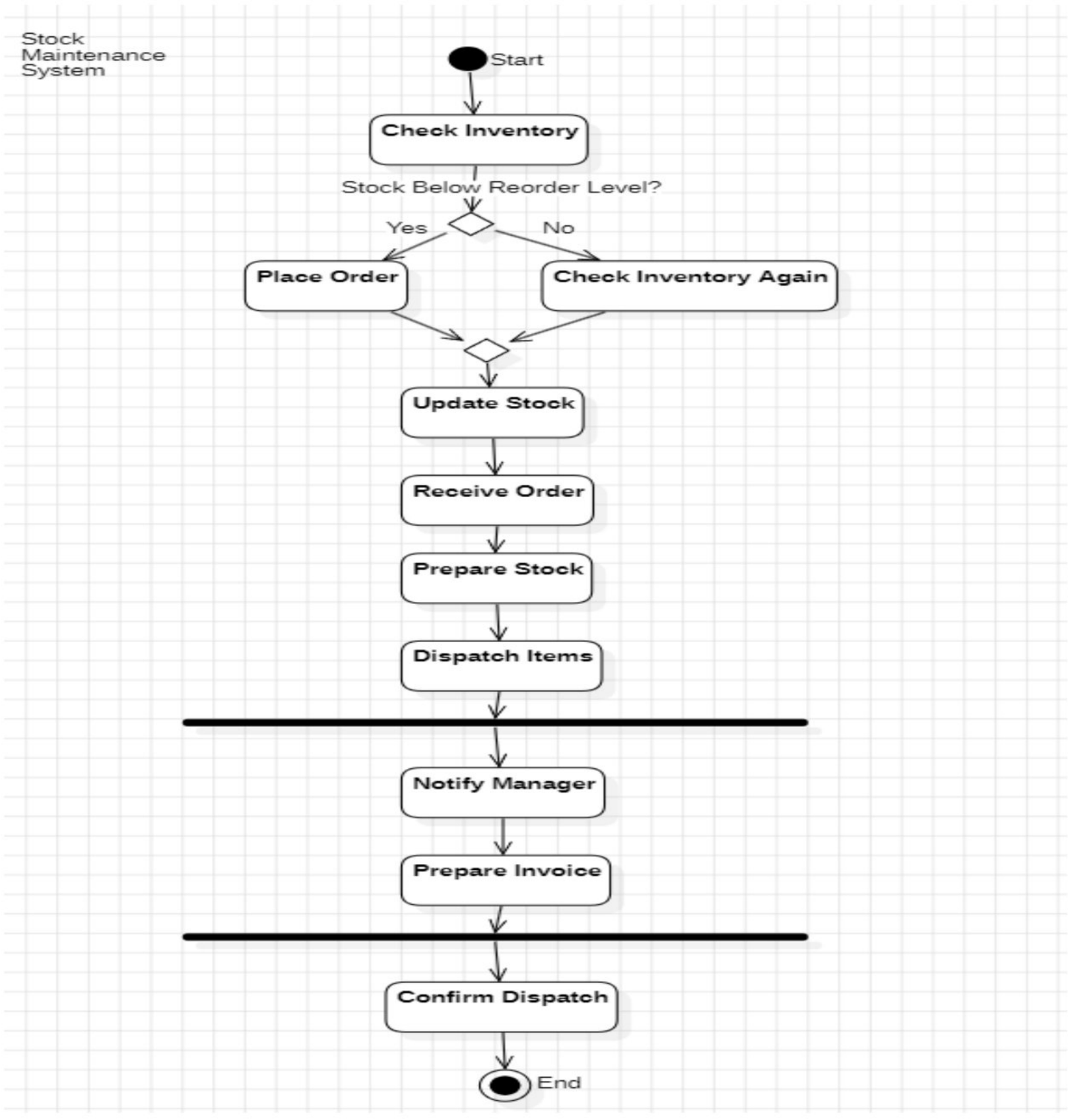


Fig 4.5 Stock Maintenance System-Activity Diagram

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.

Scanned with OKEN Scanner

5. Passport Automation System

Software Requirement Specification

5 Passport Automation System

→ Introduction

- Purpose of the Document :- The document is to outline requirements for the development of a Passport system. It ensures that it meets the operational needs of passport offices and the application experience of users.
- Scope of the Document :- It includes aspects such as application submission, status tracking, payment processing, and administrative functionalities.
- Overview :- This document gives the summary of the functional, non-functional, performance and interface requirements.

Scanned with OKEN Scanner

→ General Description :- The system is designed to streamline the passport application, processing and issuance process. It will provide a user-interface for applicants to submit their applications online, track their status, and receive notifications.

→ Function Requirements :-

* Application submission :- The system must allow users to fill out and submit passport applications online.
* Status Tracking :- It should enable applicants to track the status of their applications real-time
* Payment Processing :- The system must support online payments for application fees through various payment gateways.

→ Interface Requirements :-

* User - Interface :- The system must provide a web-based application portal for applicants.
* Database Interface :- It should interface with a RDBMS to store and retrieve data.
* Payment Gateway Integration :- The system must support integration with external payment gateways.

→ Performance Requirements :-

* The system should complete action like application submission and status checking. Also application processing update should occur within 1-2 seconds.
The system should handle upto 100,000 applicants during peak seasons without performance issues.

→ Design constraints :-

* The system must work on multiple web browsers and be mobile-responsive.
* It should use a relational database to manage applicant data and application records.
* The system should integrate with existing governmental databases for identity verfication and background checks.

→ Non-Functional Requirements :-

* Performance :- The system should respond to user actions within 3 seconds and handle upto 500 concurrent users.
* Security :- The system must implement strong security measures, including data encryption and secure access controls, to protect sensitive personal information.
* Scalability :- It should be designed to scale to accomodate increased application volumes during peak periods.

→ Preliminary Schedule and Budget :- The schedule includes the Requirement Analysis (3weeks), Design (4 weeks), Development (12 weeks), Testing (6 weeks) and Deployment and Training (2 weeks).
Budget includes Development costs ($70,000), Hardware ($10,000), Software Licenses ($5,000), Training ($3,000) and Maintenance ($12,000).

Class Diagram

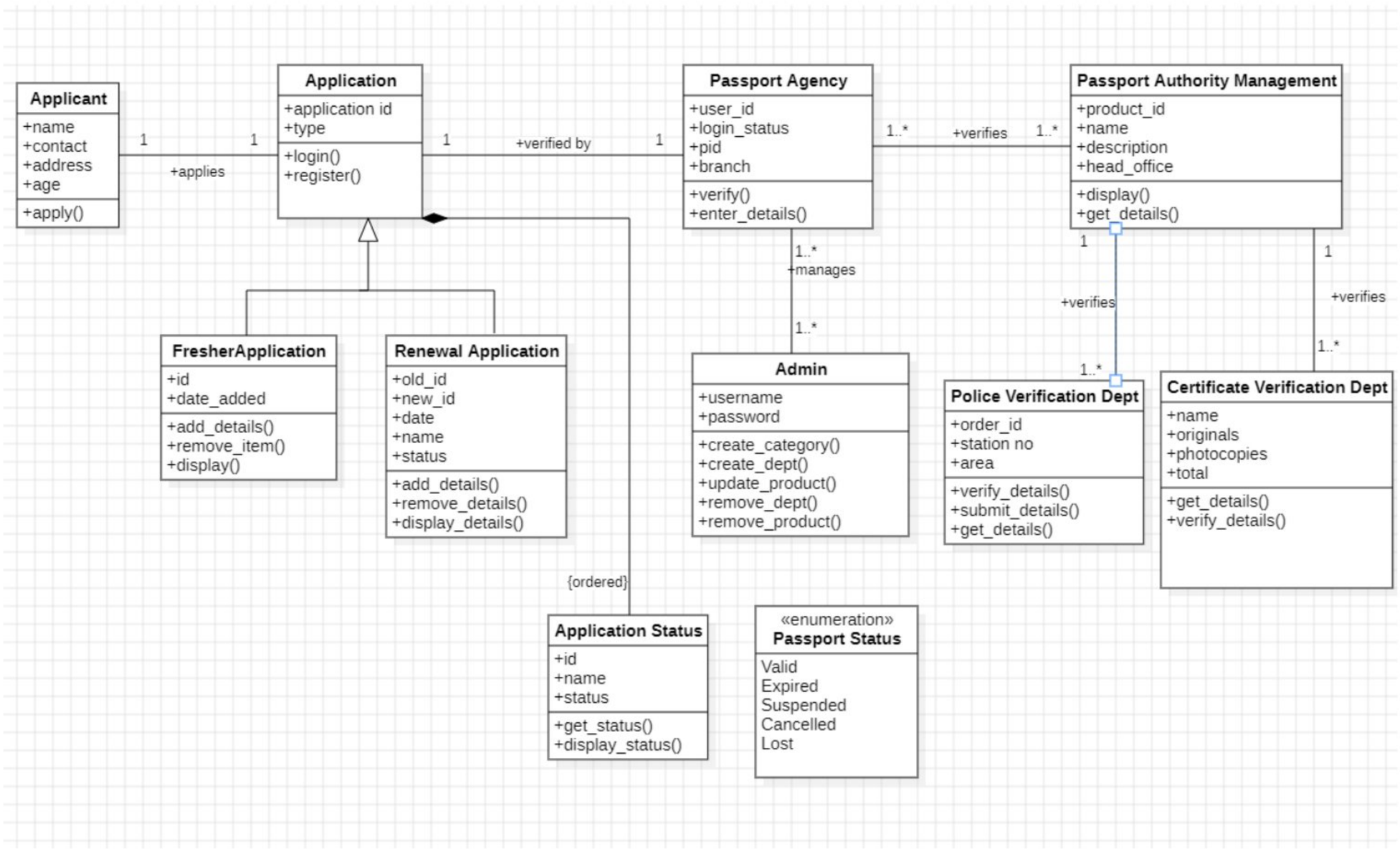


Fig 5.1 Passport Automation System - Class Diagram

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

Scanned with OKEN Scanner

State Diagram

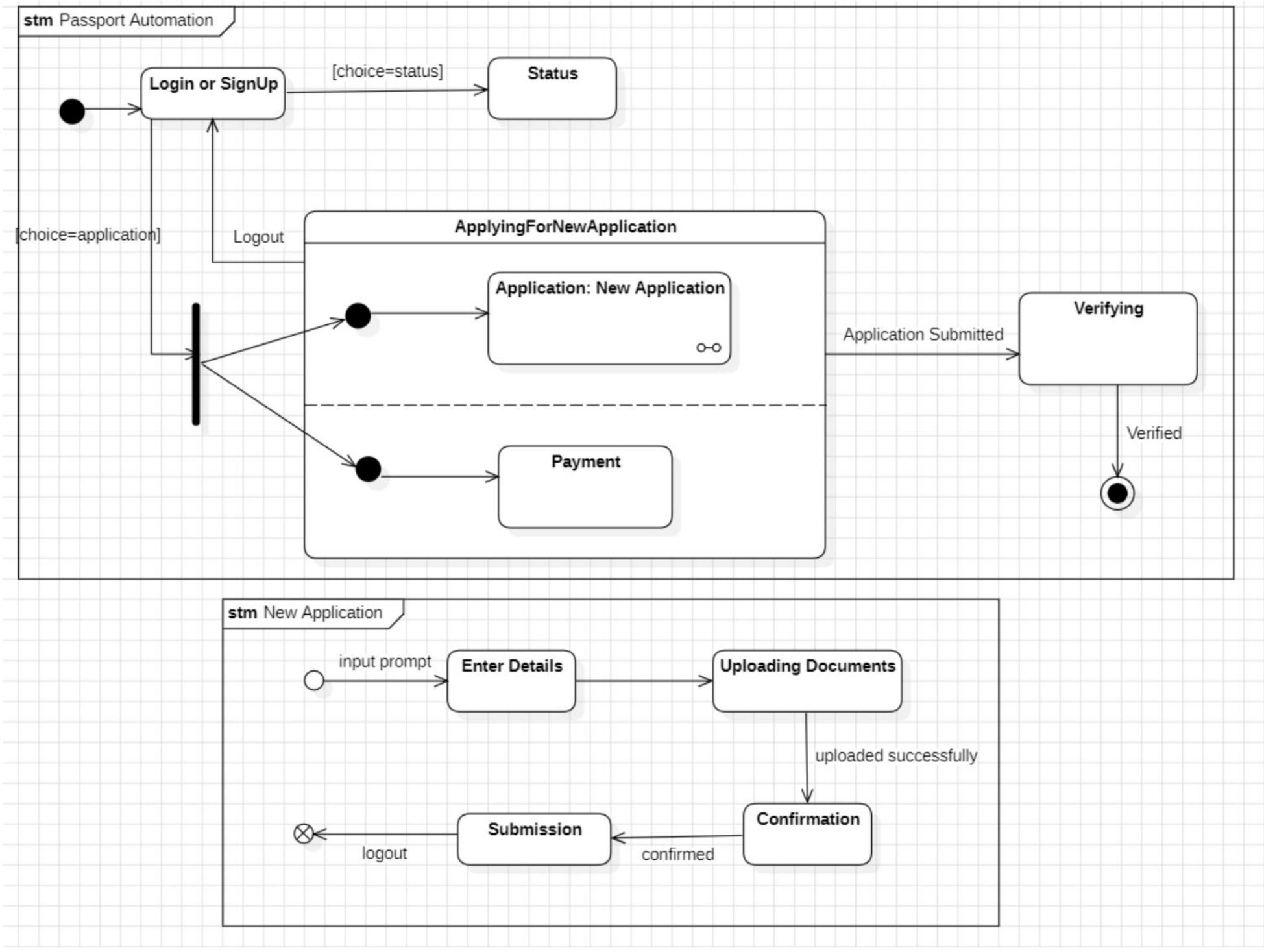


Fig 5.2 Passport Automation System - State Diagram

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.

Scanned with OKEN Scanner

Use Case Diagram

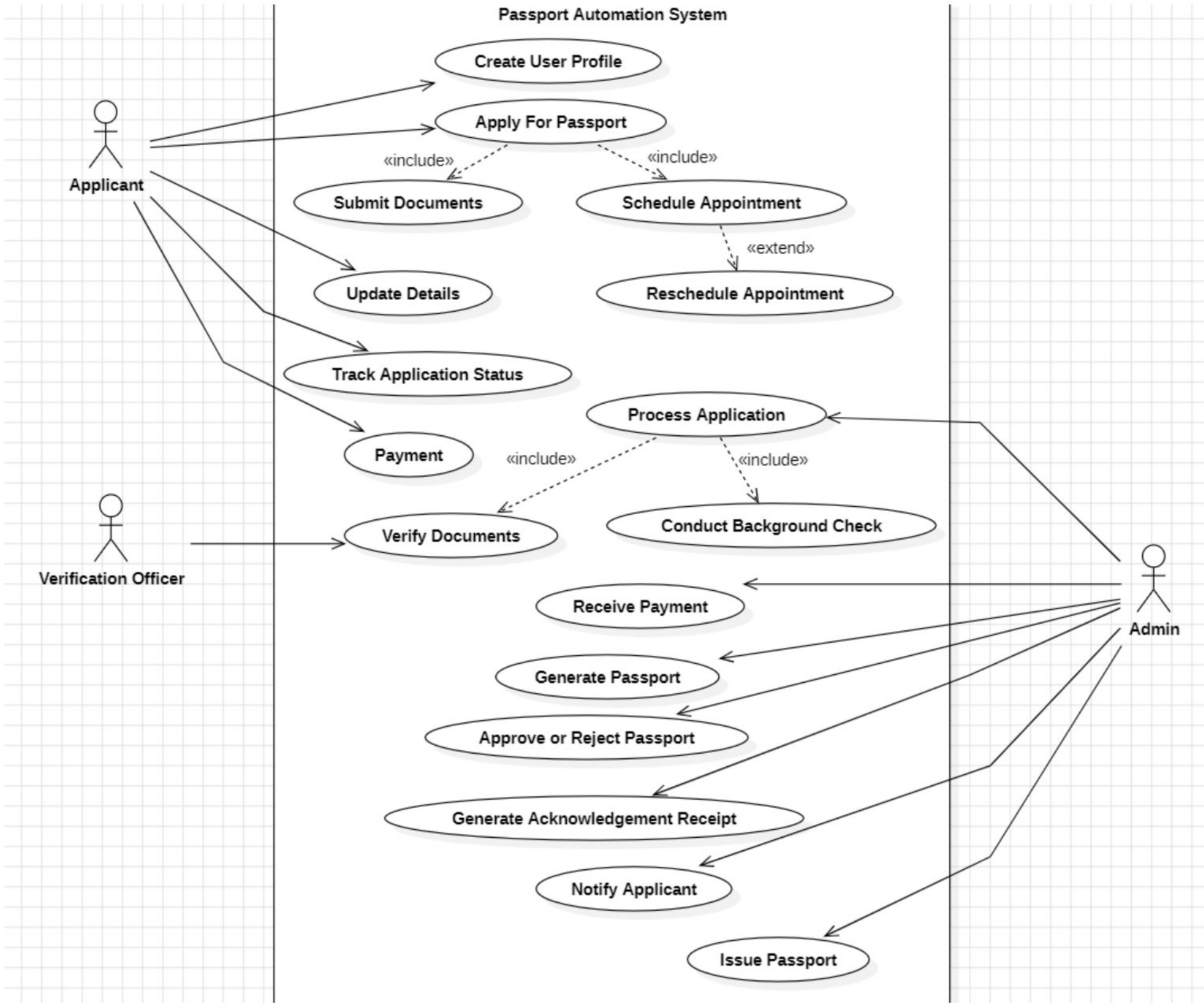


Fig 5.3 Passport Automation System - Use Case Diagram

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

Scanned with OKEN Scanner

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

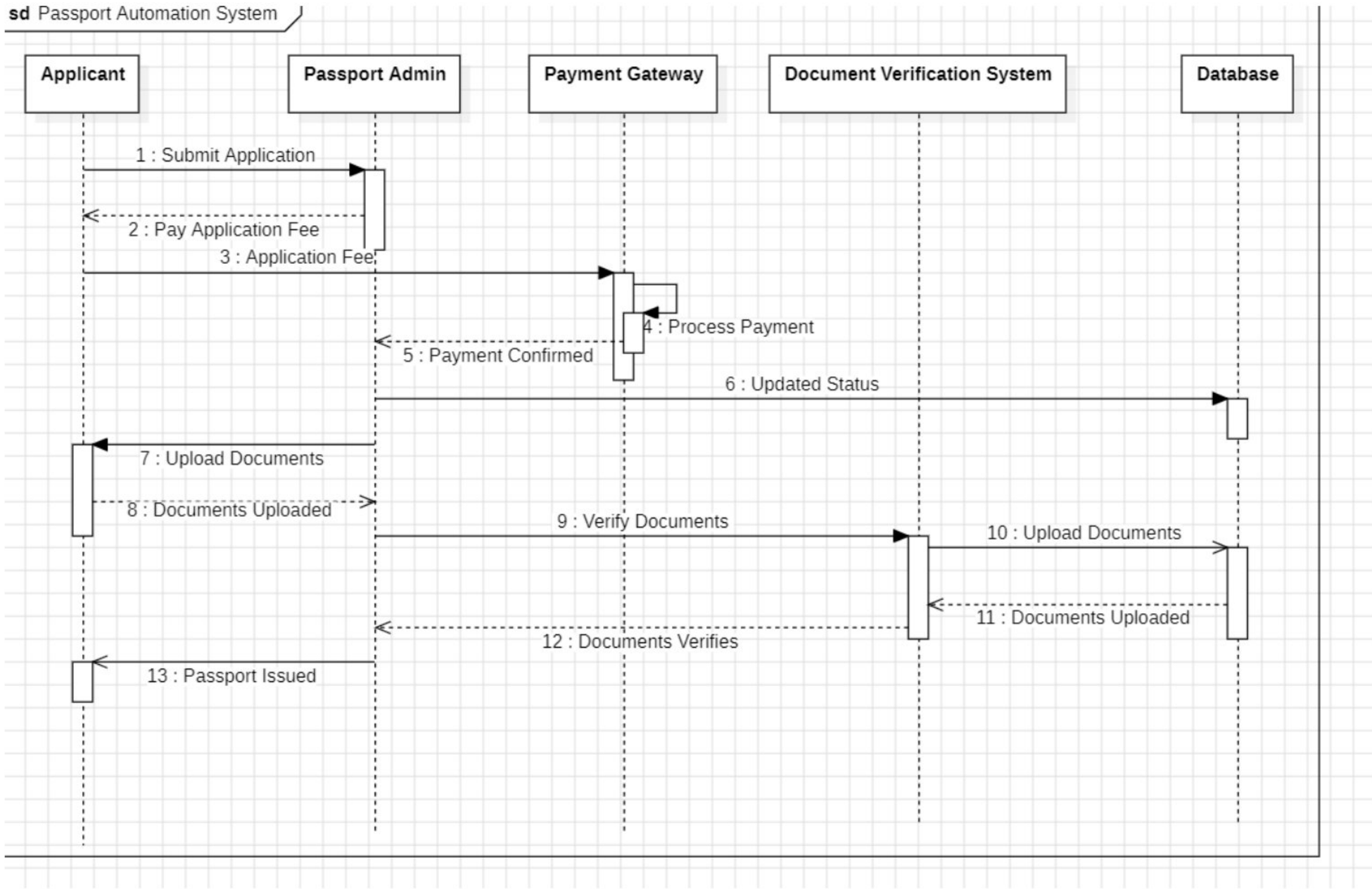


Fig 5.4 Passport Automation System - Sequence Diagram

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.

Scanned with OKEN Scanner

Activity Diagram

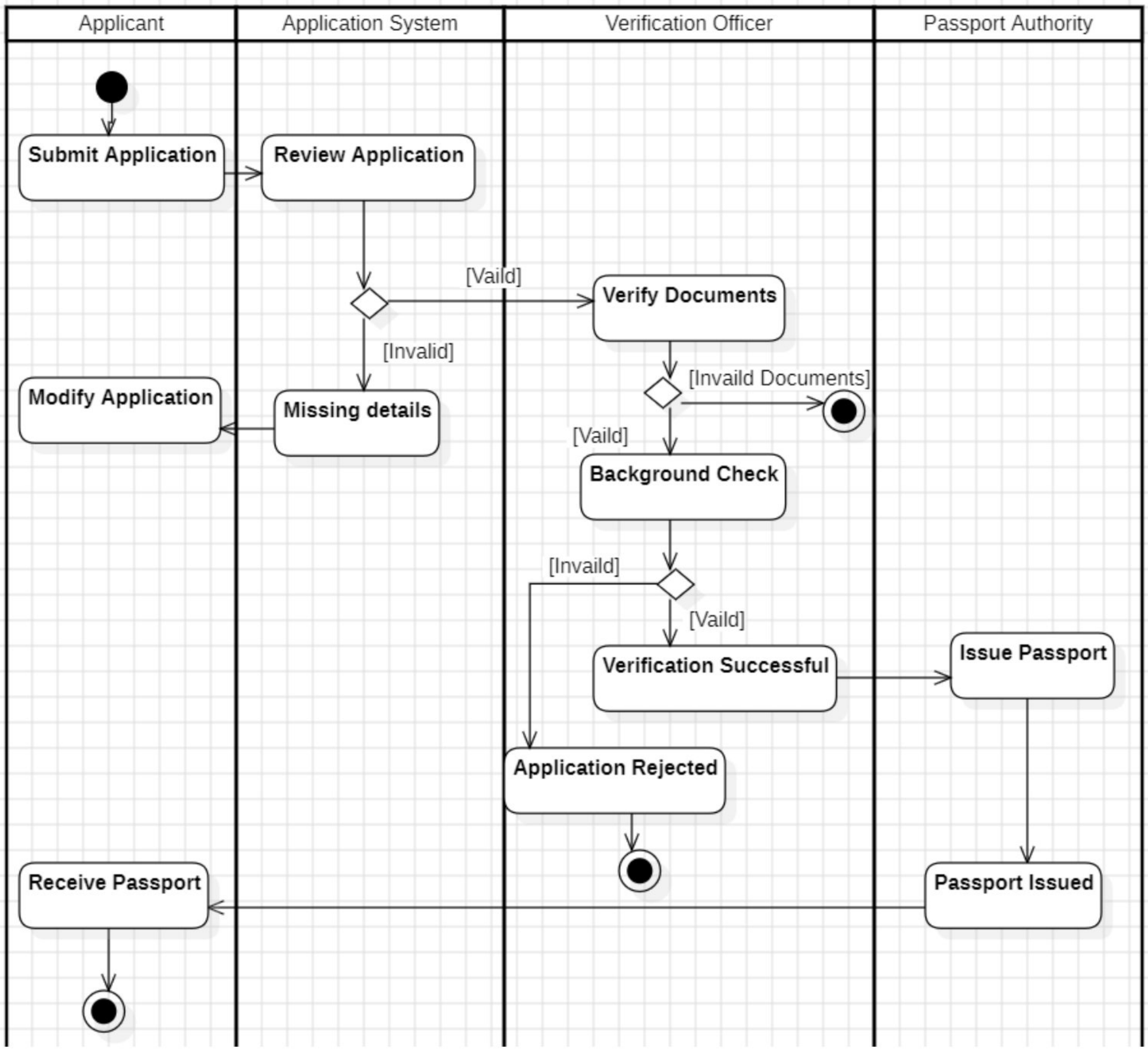


Fig 5.5 Passport Automation System - Activity Diagram

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

Scanned with OKEN Scanner

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.

Scanned with OKEN Scanner