

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:

SAMARTH KUMAR DUBEY

1BM22CS244

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

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



CERTIFICATE

This is to certify that the Object-Oriented Modelling (23CS5PCOOM) laboratory has been carried out by **SAMARTH KUMAR DUBEY**(1BM22CS244) during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

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

Table of Contents

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

1. Hotel Management System

Software Requirement Specification

<u>SRS for Hotel management</u>	
1	<u>Introduction</u>
1.1	<u>Purpose</u> : To provide clear understanding and serve as a guide during development of a Hotel management system.
1.2	<u>Scope</u> : To provide details about project development cost and time and outline all requirements for a Hotel management system.
1.3	<u>Overview</u> : The system should be able to make, cancel, generate and delete reservations, should keep track of expenses and assign staff duties.
2	<u>General Description</u>
For a Hotel management system, the functions include making and managing reservations, keeping track of expenses, supplies and checkins check-ins and check outs. It should also be able to assist user in assigning roles to the staff.	6
3	<u>Functional Requirements</u>
	<u>Management</u>
	<ul style="list-style-type: none">- Keep track of all employees, including their position and salaries- Keep track of hotel's supplies and resources

- Manage reservations and transactions of customers.

4 Interface requirements

- Software should have a clean dashboard to allow managers easy access to ~~the~~ services.
- Software should interface with online portal ~~to~~ and keep a record of transactions.
- Employees should have the ability to check their shift, tasks etc. through the system.

5 Design constraints

- Must comply to data protection laws to safeguard guests' data.
- Must be set up to be integrated with third party services like payment gateways.

6 Performance requirements

- The software should be responsive and process requests fast to ensure smooth experience.
- The software should have good uptime, providing 24/7 service.
- The software should be able to handle multiple requests at once.

Date _____ / _____ / _____

7 Non-functional Attributes

- i) Usability \Rightarrow easy to use
- ii) Reliability \Rightarrow System should be dependable such that critical systems do not fail
- iii) Security \Rightarrow Software should protect potentially sensitive user data

89 Preliminary schedule and budget

Estimated development time for the system is 7 months with an approximate budget of 70,000 US\$.

Class Diagram

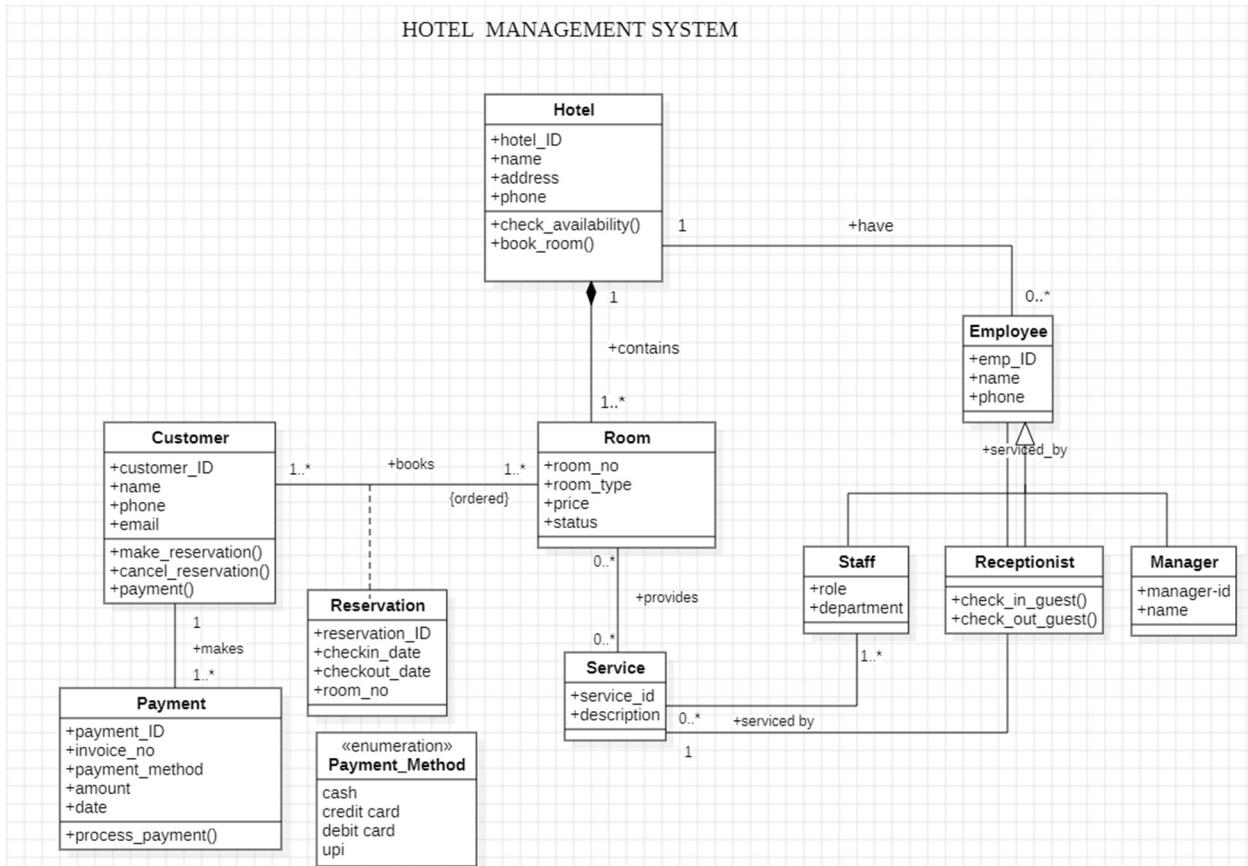


Fig1.1 Hotel Management System - Class Diagram

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

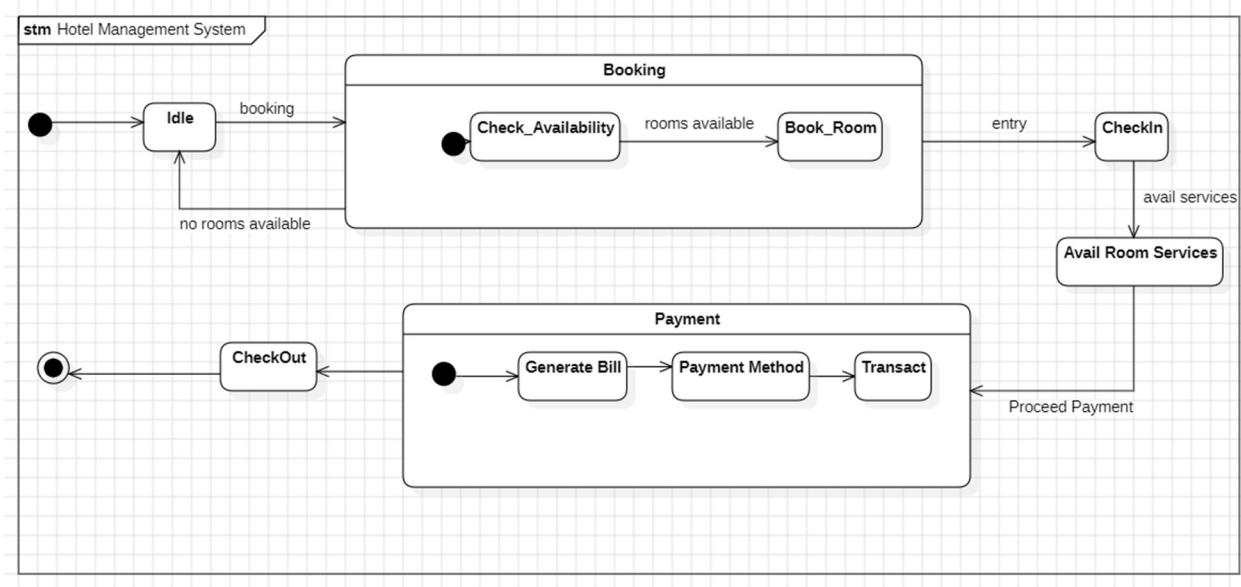


Fig1.2 Hotel Management System - State Diagram

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

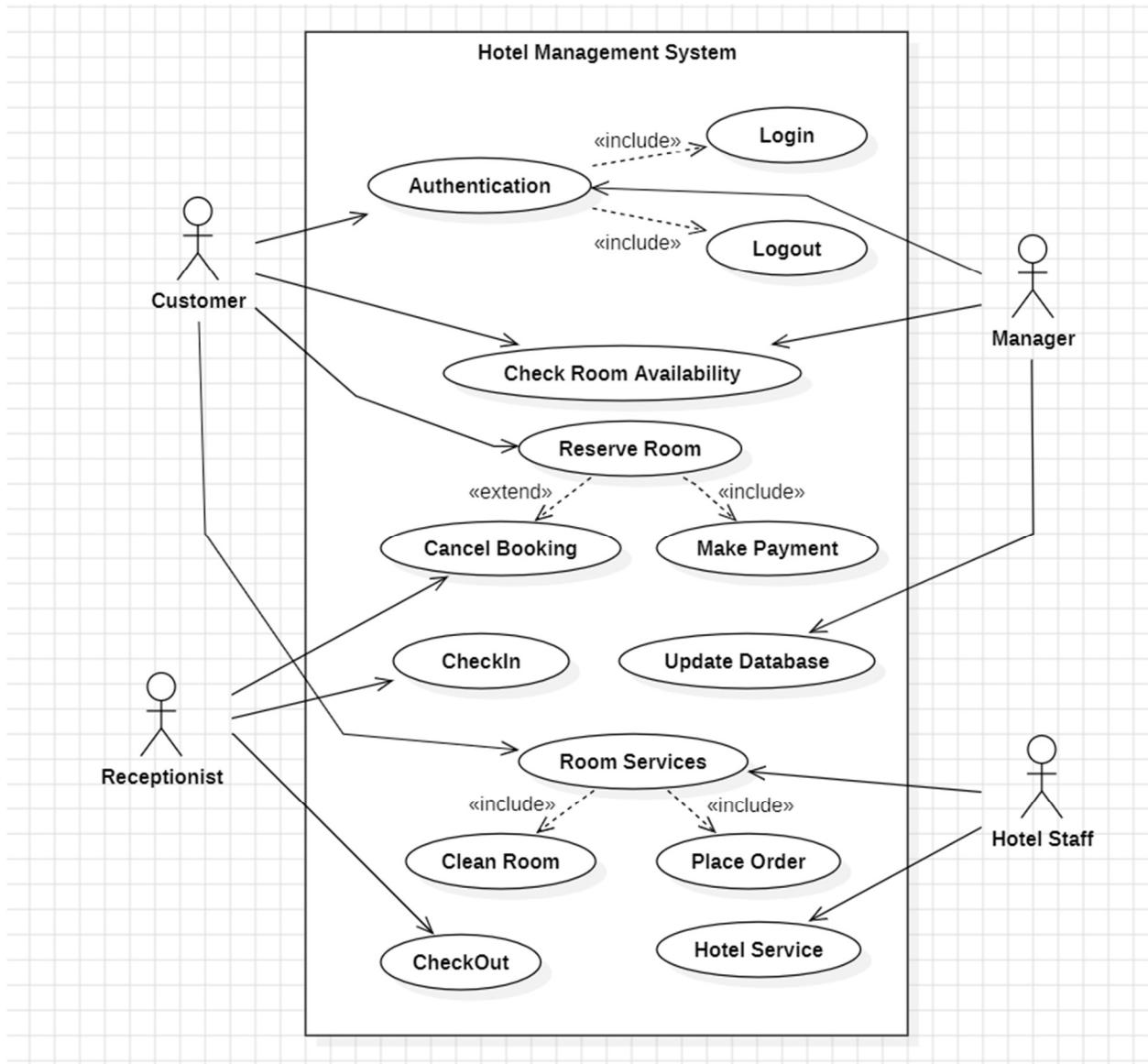


Fig1.3 Hotel Management System - Use Case Diagram

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

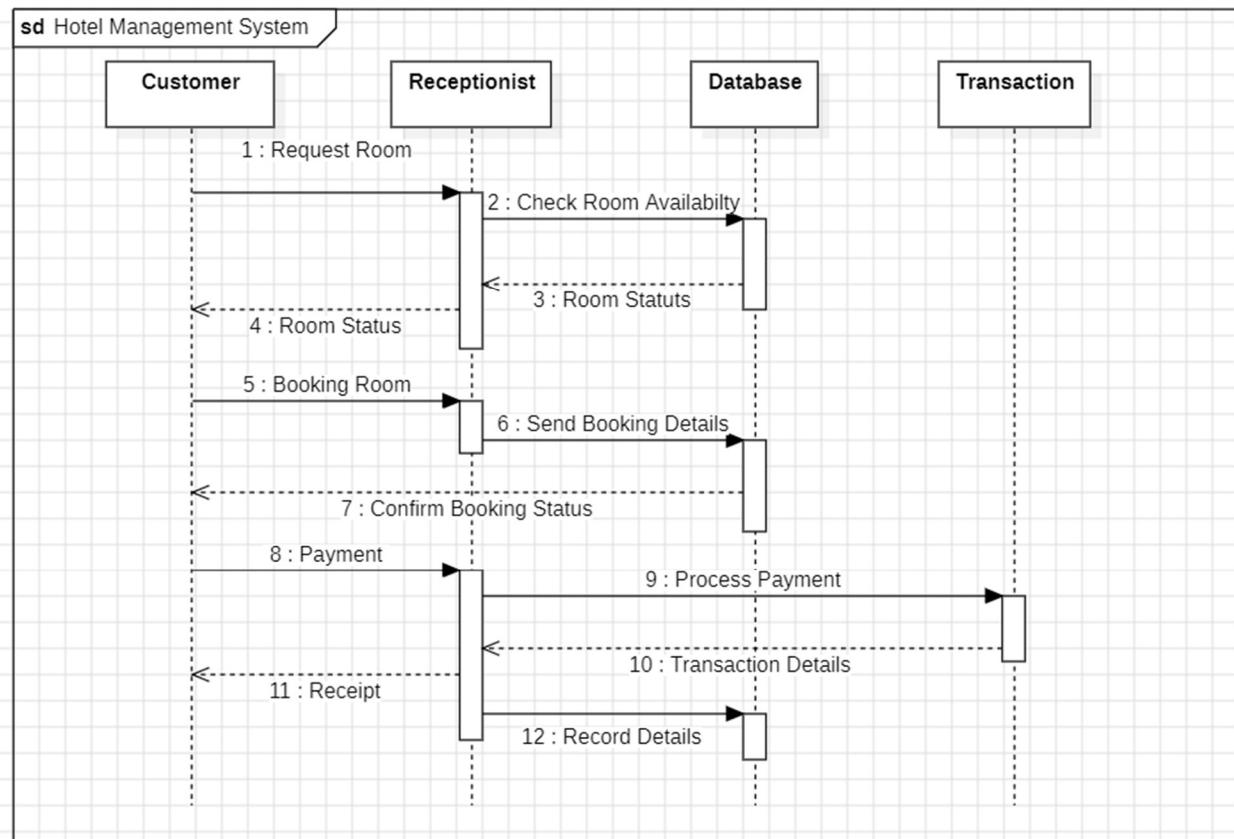


Fig1.4 Hotel Management System - Sequence Diagram

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

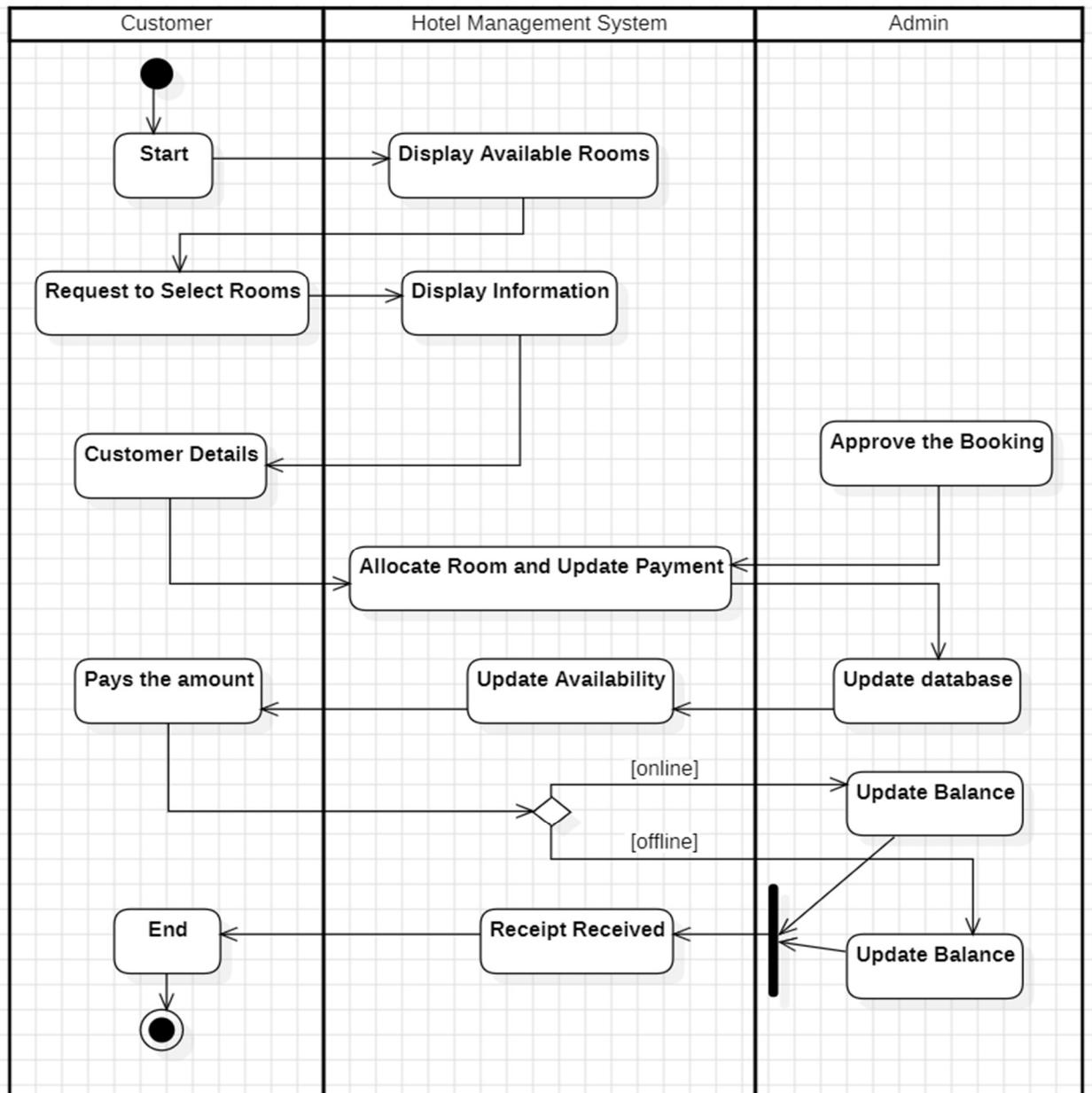


Fig1.5 Hotel Management System - Activity Diagram

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

II a ~~Railway Reservation system~~
~~Credit card System~~

1 Introduction

1.1 Purpose :

This document aims to highlight functionalities, requirements and other non functional attributes of a Railway Reservation system to serve as a guide during development cycle.

1.2 Scope

The document covers the functional and technical aspects of a railway reservation system. It includes features like ticket booking, cancelling, schedule viewing.

1.3 Overview

The system includes a facility to ~~cancel~~, reservations, reserve seats, view train schedules,

2 General Description

There will be two categories for users, staff and customers.

Customers should be able to make reservations based on different parameters like duty pass

Jatka etc., as well as check booking status, train schedule, seating arrangement.

staff should be able to view records and answer service calls.

3 Functional requirements

- users should be able to reserve and cancel seats.
- user should be able to view train schedule, running status, seat arrangement etc..
- user should be able to request cleaning and related services.

4 Interface requirements

User Interface

- should be simple and intuitive
- Should include ^{local} language support for easy usage

Database Interface

- software should update in real time the the reservation status of users
- software should also update

Payment Interface

- should integrate a third party service for transactions

Date / /

5 Performance Requirements

- The system should be able to handle large amount of simultaneous users during peak hours.
- It should have good response time, good uptime and good scalability.

6 Design constraints

- System should be compatible with commonly used platforms like web and mobile phones.
- System should adhere to industry protocols regarding transaction.

7 Non functional attributes

- i) Security: The system should ensure all personal data is secure.
- ii) Portability: The system should be accessible on mobile, tablets, desktops etc.
- iii) Maintainability: The system should be designed to allow easy deployment and debugging.

8 Preliminary schedule and budget

Total development time is estimated to be 8 months and with an approximate budget of 100000 USD.

Class Diagram

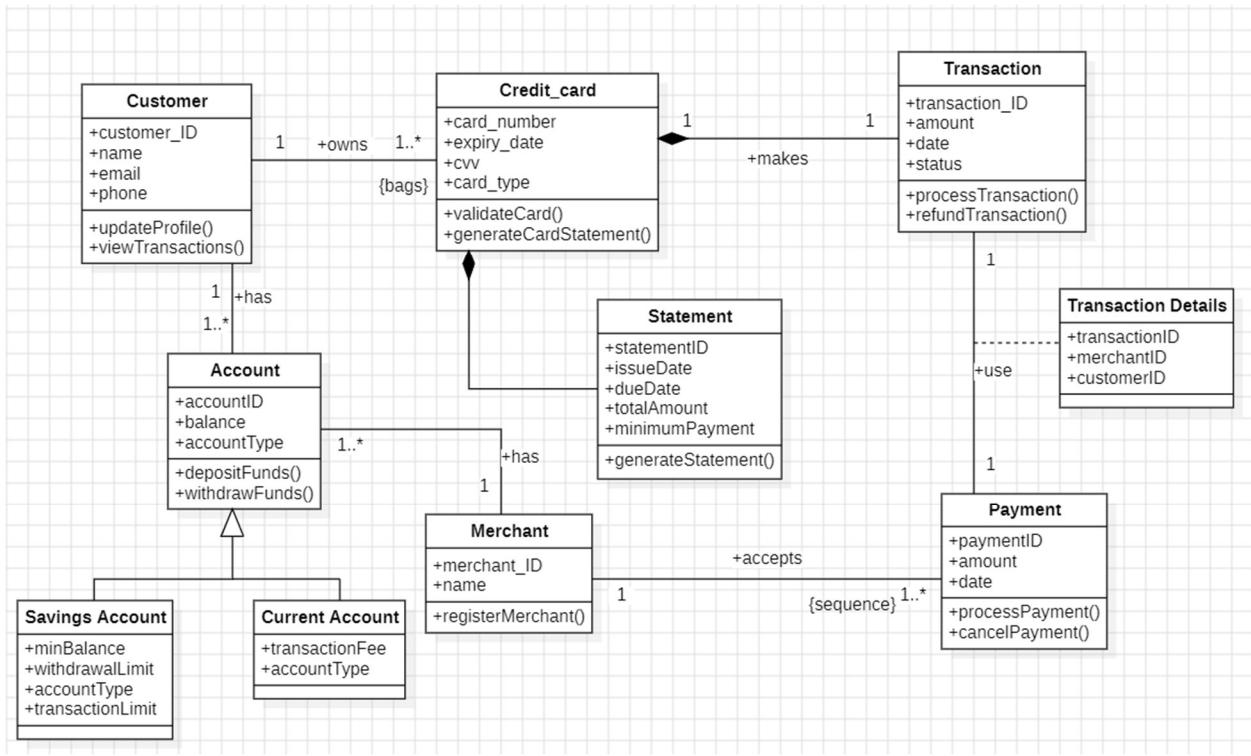


Fig 2.1 Credit Card Processing System - Class Diagram

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

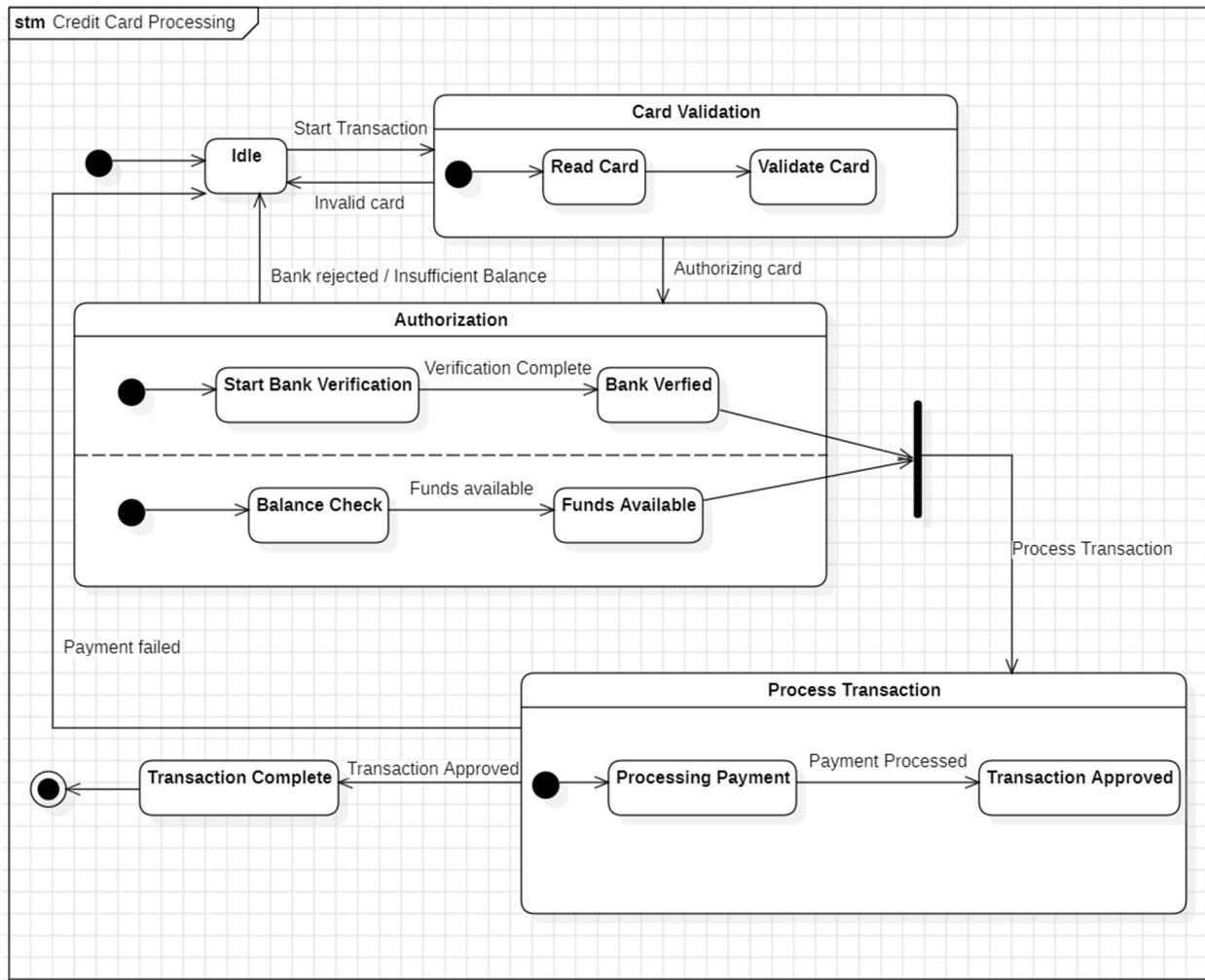


Fig 2.1 Credit Card Processing System - State Diagram

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

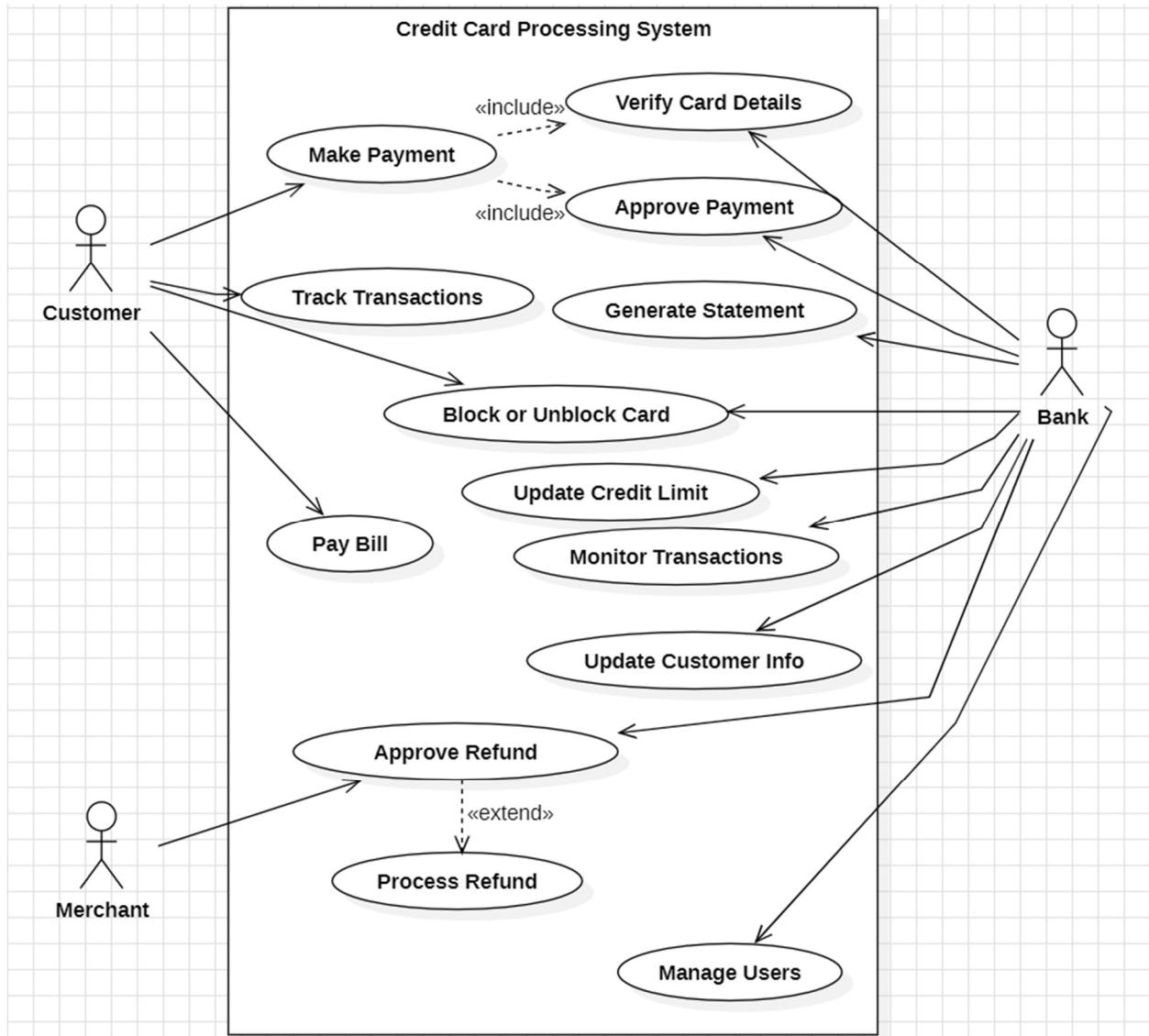


Fig 2.1 Credit Card Processing System - Use Case Diagram

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

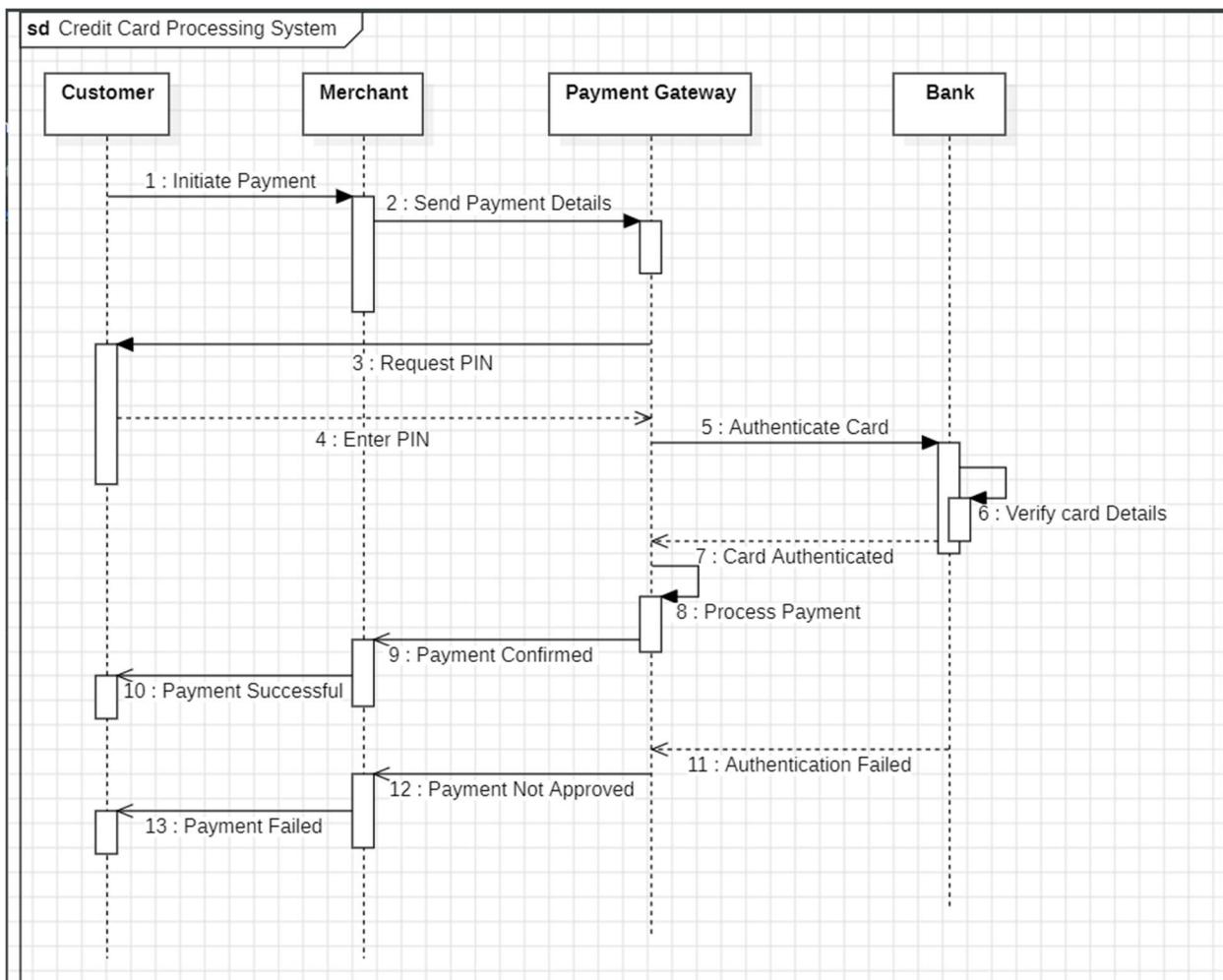


Fig 2.1 Credit Card Processing System - Sequence Diagram

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

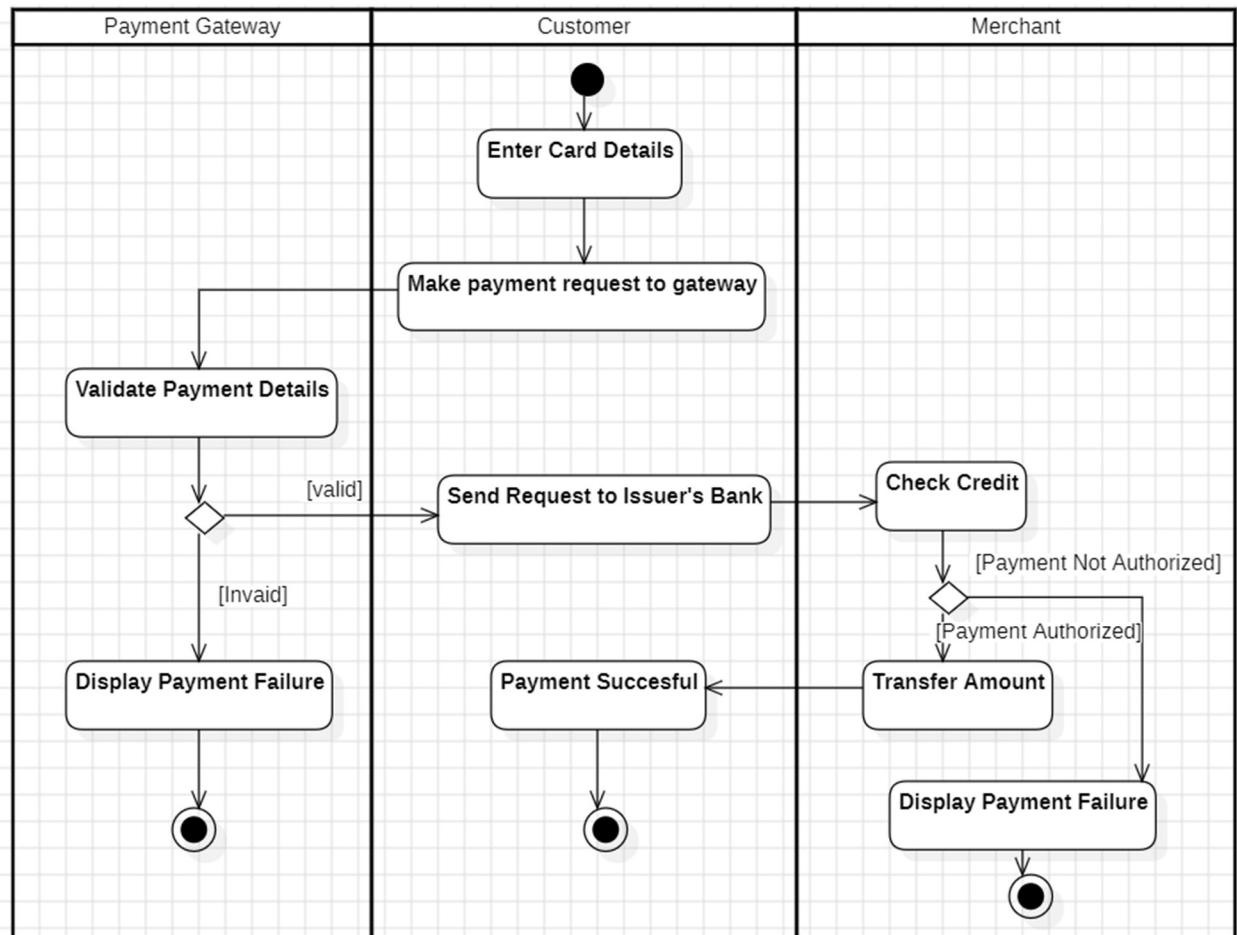


Fig 2.5 Credit Card Processing System - Activity Diagram

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

II		Library management system
1		Introduction
1.1		<u>Purpose</u>
1.1	1.1	<p>Purpose of this document is to provide a comprehensive overview of its functional and non-functional requirements for the development of an online shopping system. This document is essential for guiding the development process and ensuring system meets the needs of both customer and administrator.</p>
1.1	1.2	<u>Scope</u>
1.1	1.2	<p>The document covers the design, functionality, and operational aspects of an online shopping system. It intends to facilitate the purchase of products over the internet by providing users with a convenient interface for browsing, searching and ordering products.</p>
1.1	1.3	<u>Audience</u>
1.1	1.3	<p>The online shopping system is designed to provide customers with an efficient and user-friendly platform for purchasing goods.</p>

2 General description

The system is intended to accommodate both customers and administrators. Intended functionalities include letting users register and login, let them browse through a catalog of products, let them add/removal items from the cart, and allow admins to update and manage products, process orders and view statistics.

3 Functional Requirements

- customer registration, authentication and login
- let customers browse and search through items with filters of choice
- let users review cart, apply coupon codes and pay
- let users see order history
- let admins manage products and see statistics

4 Interface requirements

- System should have a simple web based interface for users
- There should be a separate database for admins for managing inventories, processing orders and viewing stats.
- Database should be able to interact with and store data of the screen

5 Performance requirements

- System should have a good response time and uptime to ensure good experience for user
- System should be able to handle large amount of simultaneous users

6 Design Constraints

- ~~that~~ System should be ~~able to~~ compatible with all modern browsers and mobile devices
- System should use a relational database for products.
- System should comply with relevant e-commerce laws and regulations

7 Non functional attributes

- i) Security \Rightarrow System should implement strong measures to protect customer data
- ii) Availability \Rightarrow System should be accessible on mobile devices & tablets in addition to desktop computers
- iii) Reliability \Rightarrow The system should reliably process orders and transactions.

8 Preliminary Schedule

Approximate development time is thought to be 7 months with estimate of budget of 170000 dollars.

Class Diagram

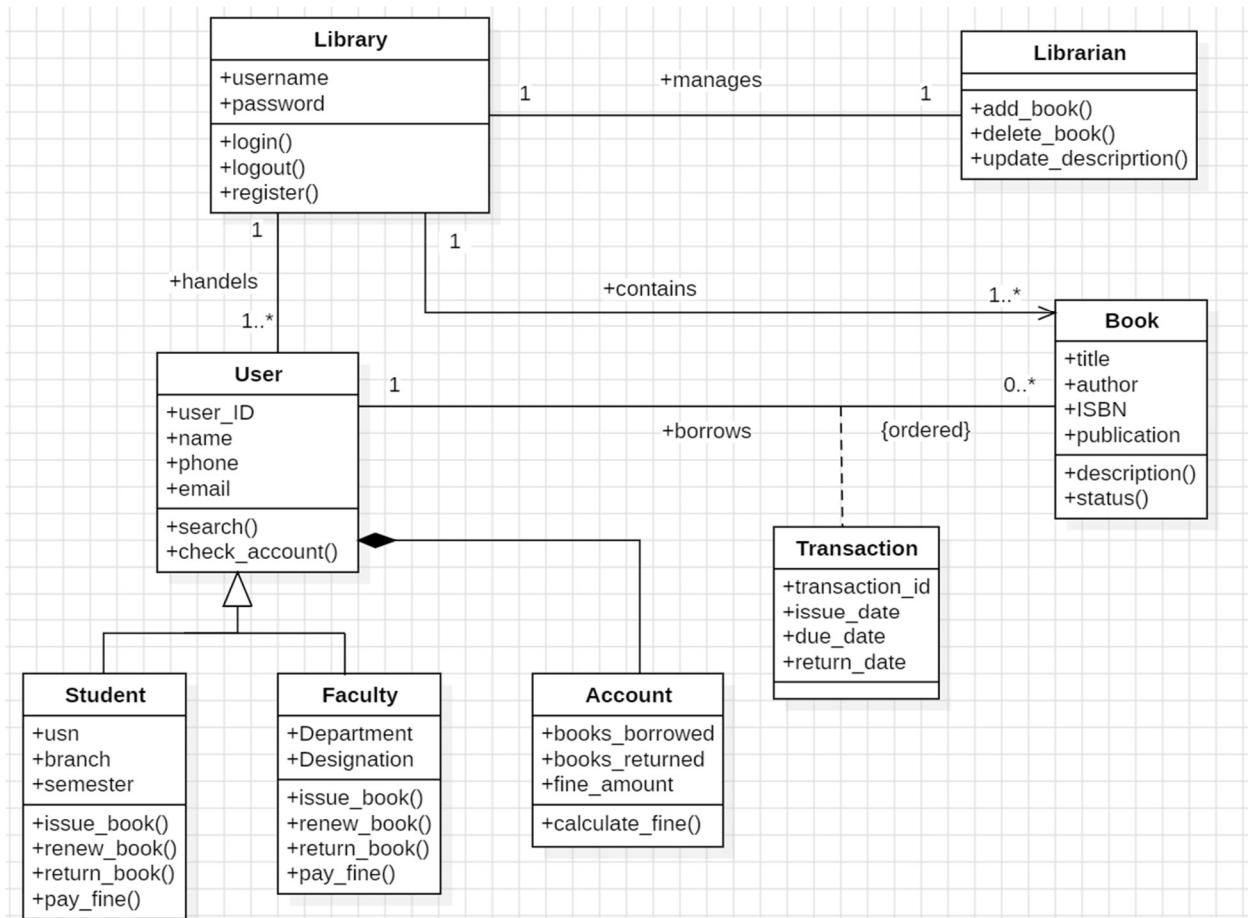


Fig 3.1 Library Management System -Class Diagram

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

State Diagram

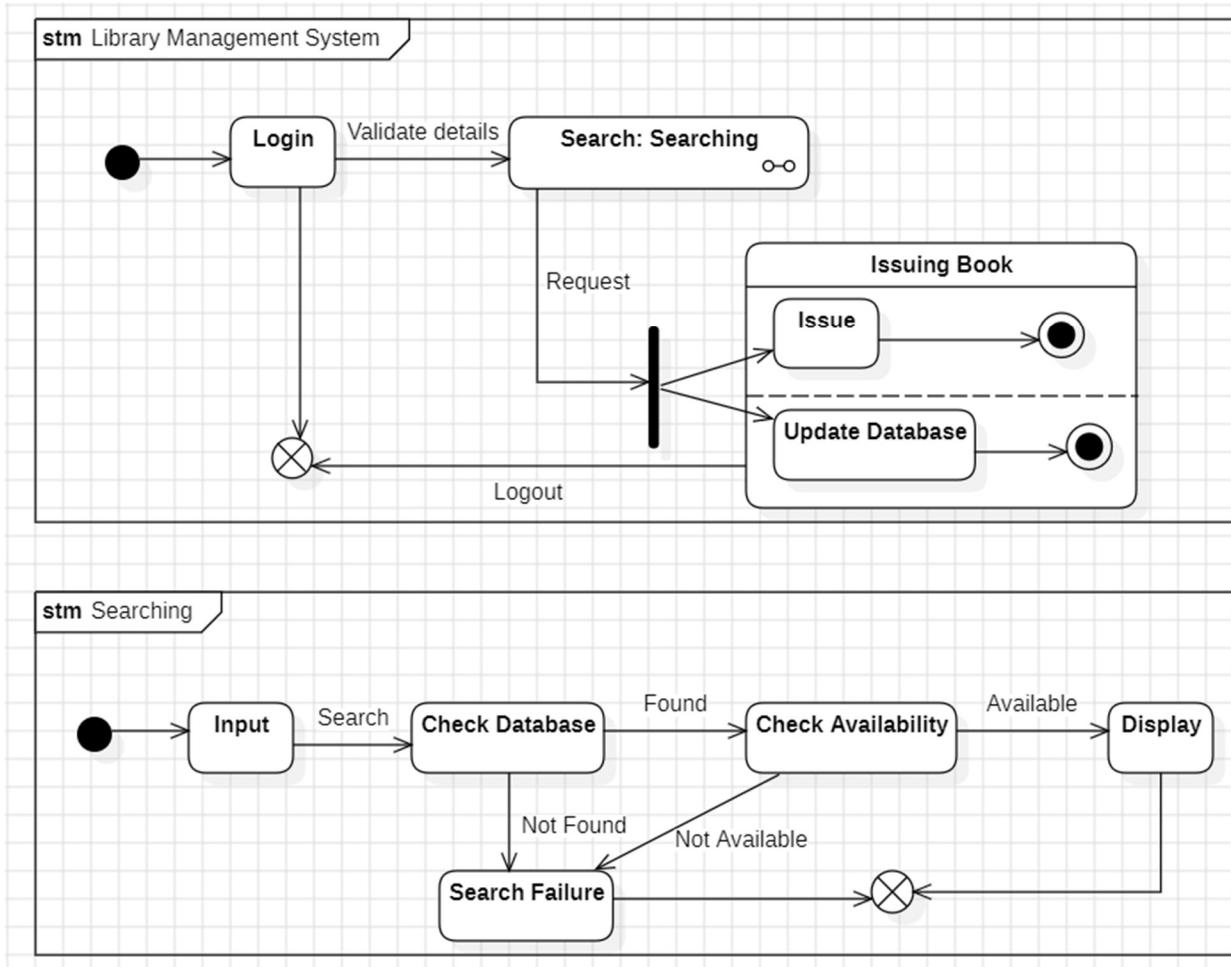


Fig 3.2 Library Management System - State Diagram

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

Use Case Diagram

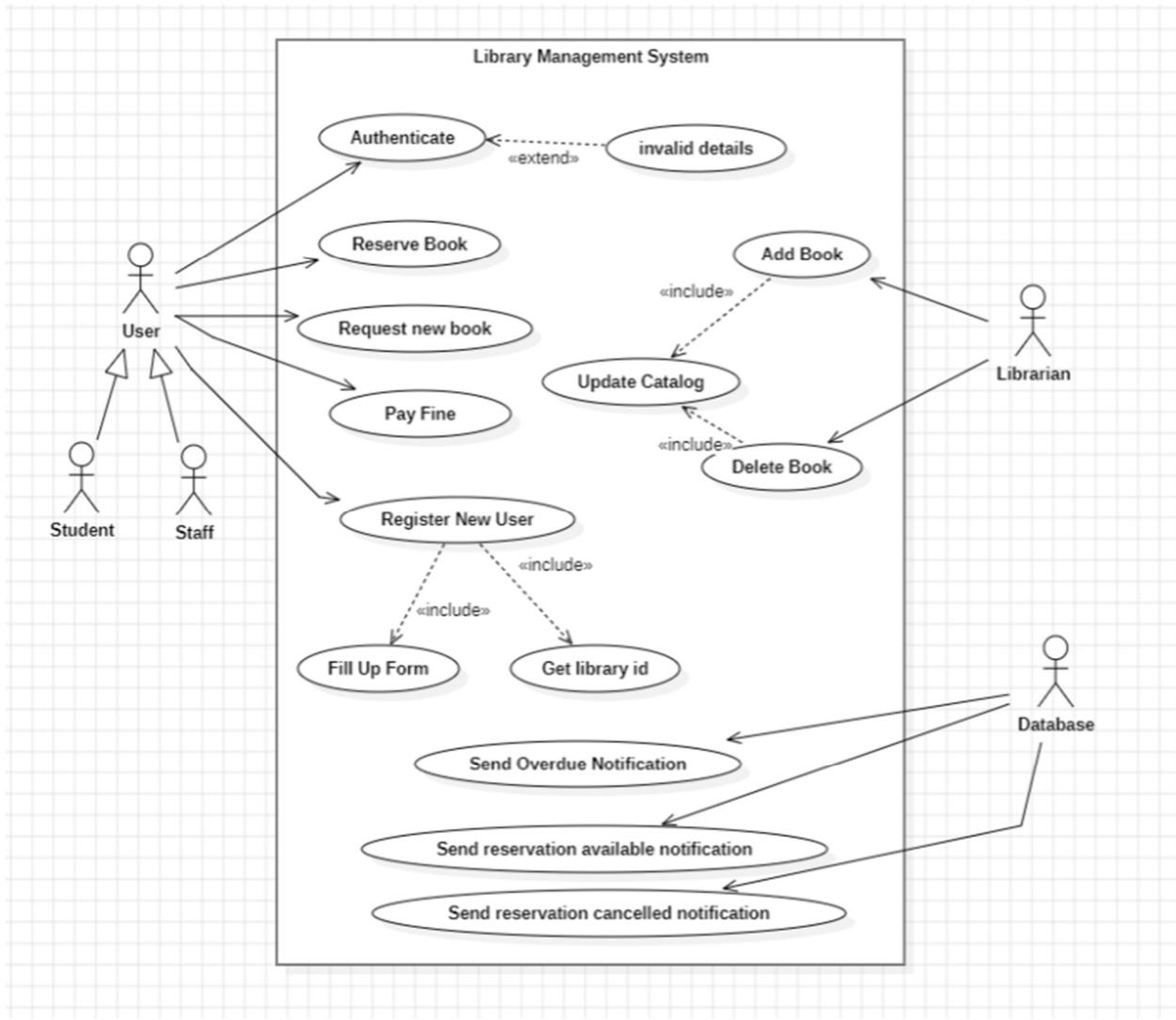


Fig 3.3 Library Management System - Use Case Diagram

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

Sequence Diagram

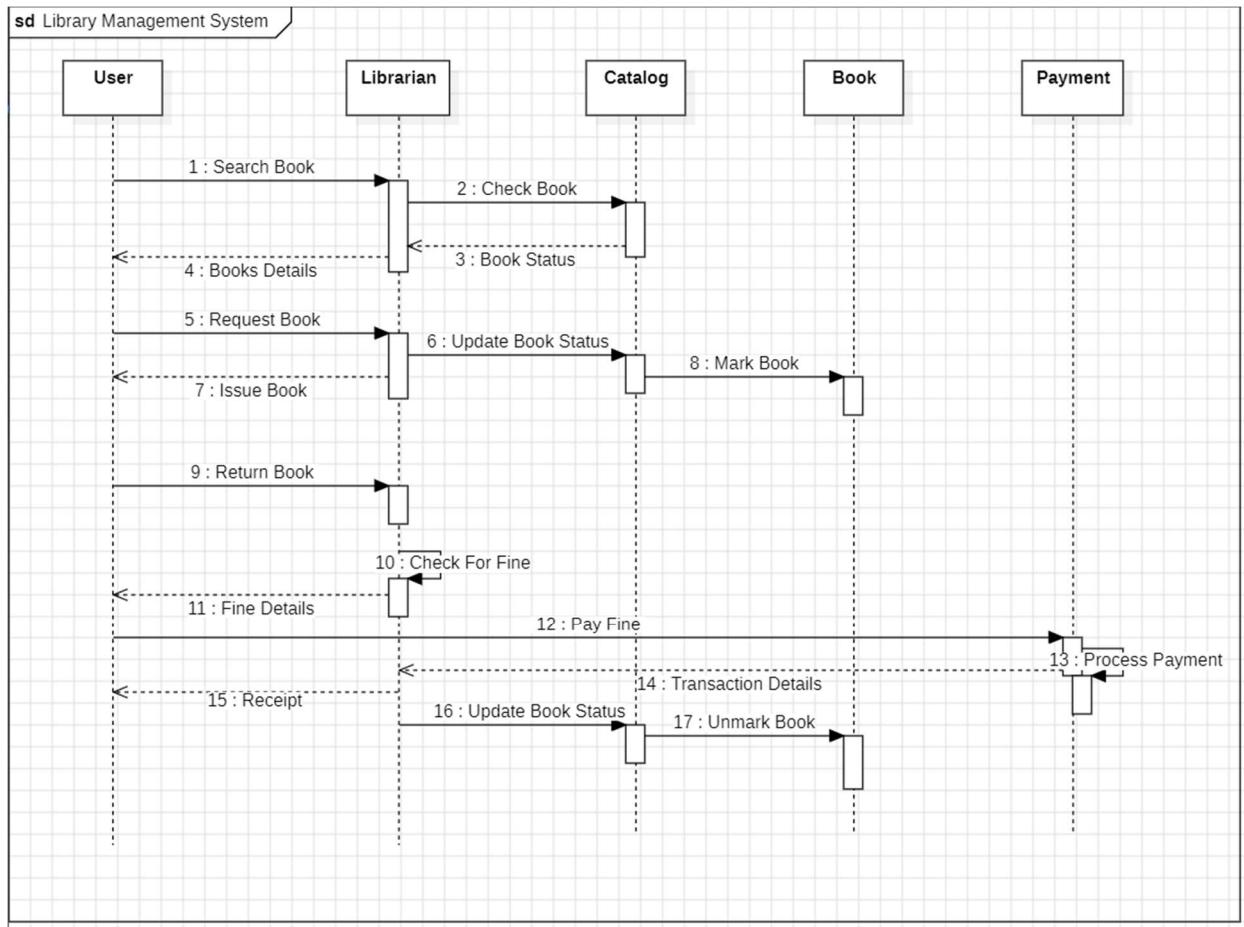


Fig 3.4 Library Management System - Sequence Diagram

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

Activity Diagram

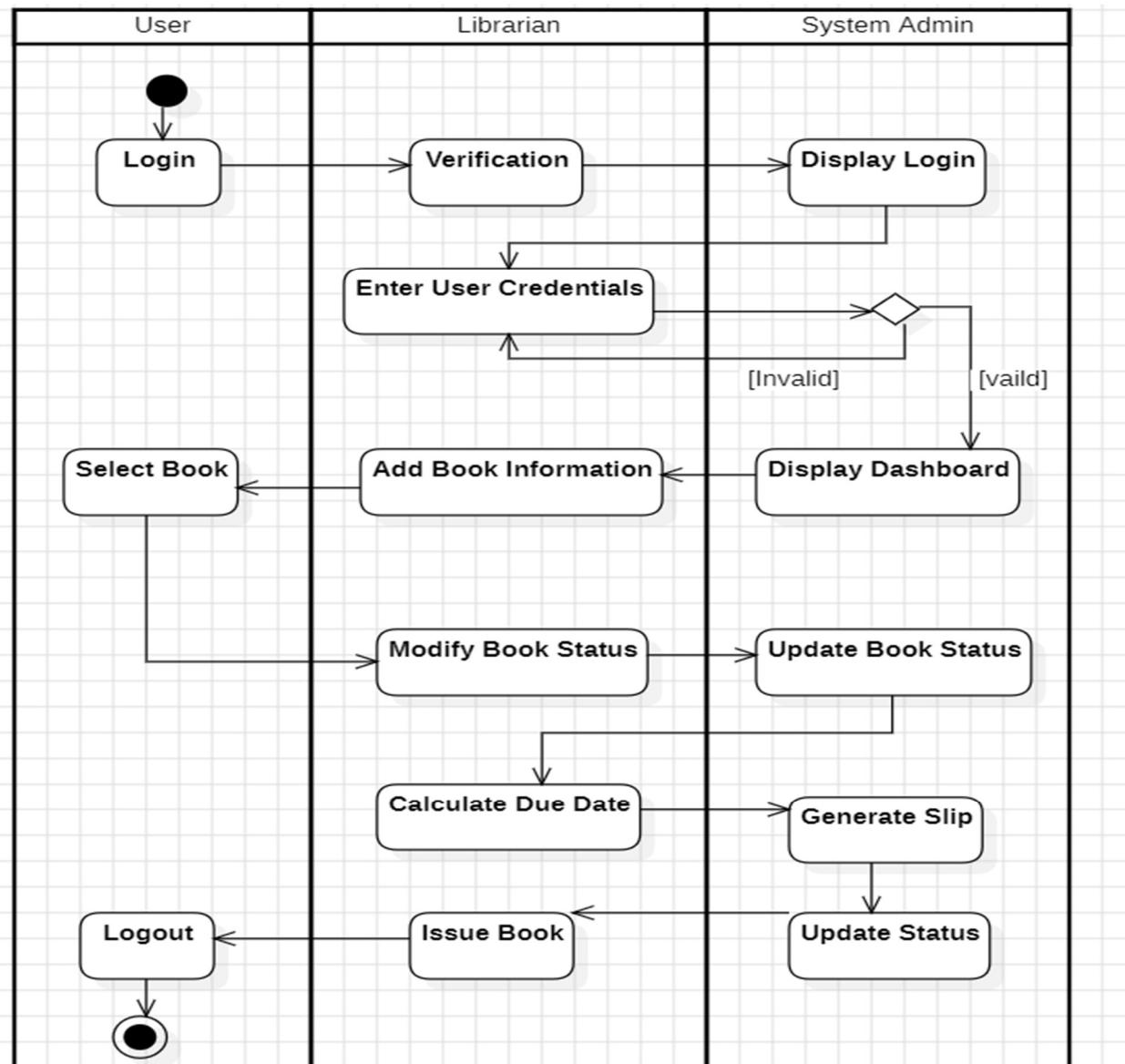


Fig 3.5 Library Management System - Activity Diagram

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

4. Stock Maintenance System

Software Requirement Specification

Saathi	
Date _____	
V	Stock maintenance system
1	<u>Introduction</u>
1.1	<u>Purpose</u> The stock maintenance system ensures that to capture the price, movement & also providing insights of the market
1.2	<u>Scope</u> The stock management defines necessary functionalities for stock maintenance system with aims to monitor stock levels, generate alerts, automate stock replenishment requests and provide real time updates.
1.3	<u>Overview</u> The stock maintenance system will serve businesses by automating inventory tracking and management. It will allow users to view current stock levels, receive low stock alerts, and track incoming and outgoing stock.
2	<u>General Description</u>
	The stock maintenance system is designed for businesses to efficiently manage their inventory. Key functions include: stock level monitoring, stock transactions, inventory replenishment and reporting.
	The system will cater to warehouse managers, inventory controllers, and business owners.

Date / /

3 Functional Requirements

- Users should be able to view current stock levels.
- Users should be able receive alerts once stocks are near depleted.
- Requests for replenishment should be automated.
- Users should receive reports of incoming and outgoing stocks.

7

4 Interface Requirements

- user interface should be a simple web based interface to allow user easy interaction.
- System should interact with a central database to store, retrieve and process information about tasks.

8

5 Performance Requirements

- System ~~should~~ should have minimal response time.
- System should be able to handle large scale businesses with ~~more~~ thousands of users.
- System should have 99% uptime.

6 Design Constraints

- System should be able to handle ~~large~~ bulk amount of data.
- System should adhere to rules and regulations.

Page No. []

of the market.

- Should Integrate with existing finance systems

7 Non functional attributes

- Security: System should implement ~~to~~ encryption protocols to ~~allow~~ safeguard ~~to~~ sensitive information.

- Usability: Ensuring an easy to navigate interface for non technical staff

- Reliability: System should have an uptime of 99% and must be ~~be~~ operational quickly if ~~the~~ its down due to errors

8 Preliminary schedule & budget

Estimated development time is 9 months
with 15000 USD being the estimated budget.

Class Diagram

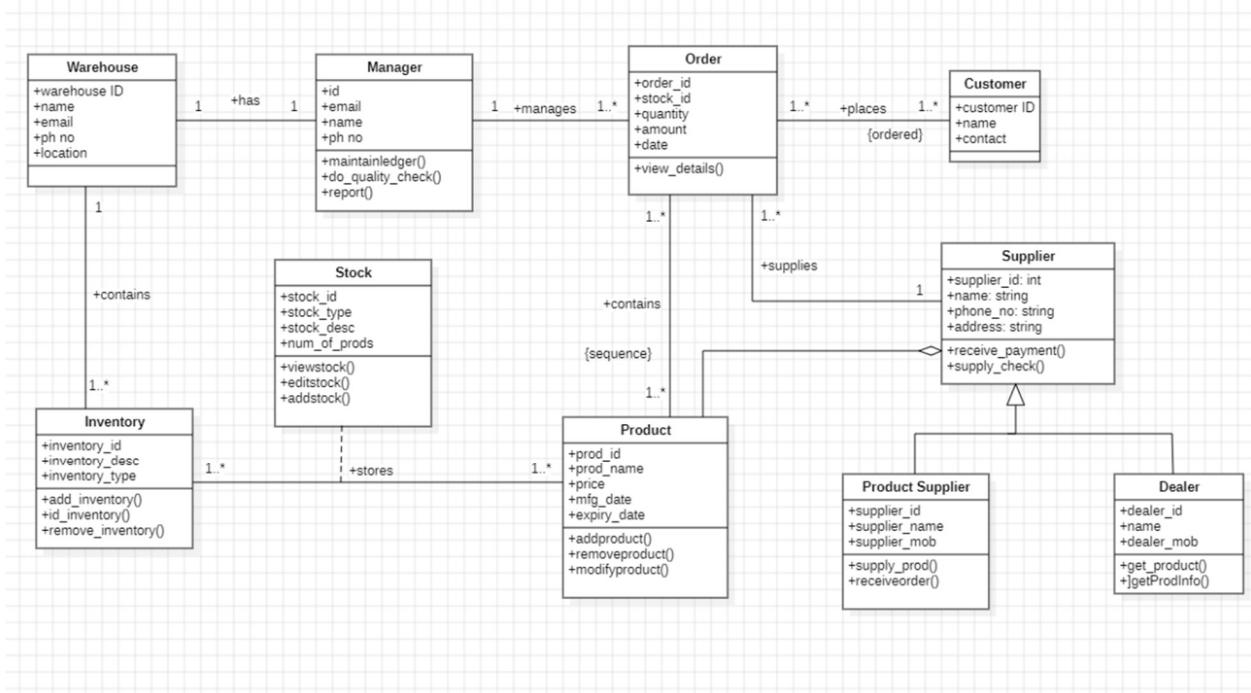


Fig 4.1 Stock Maintenance System - Class Diagram

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

State Diagram

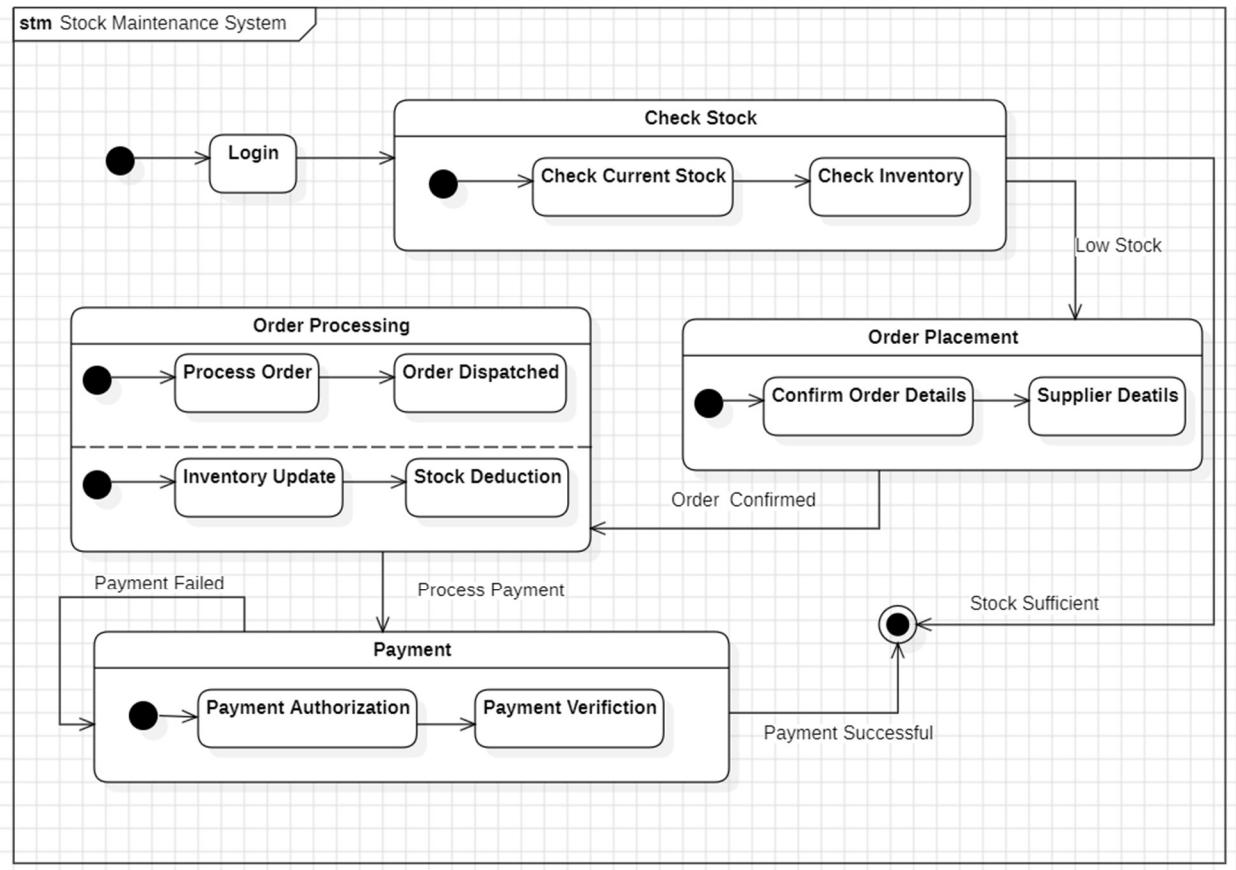


Fig 4.2 Stock Maintenance System - State Diagram

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

Use Case Diagram

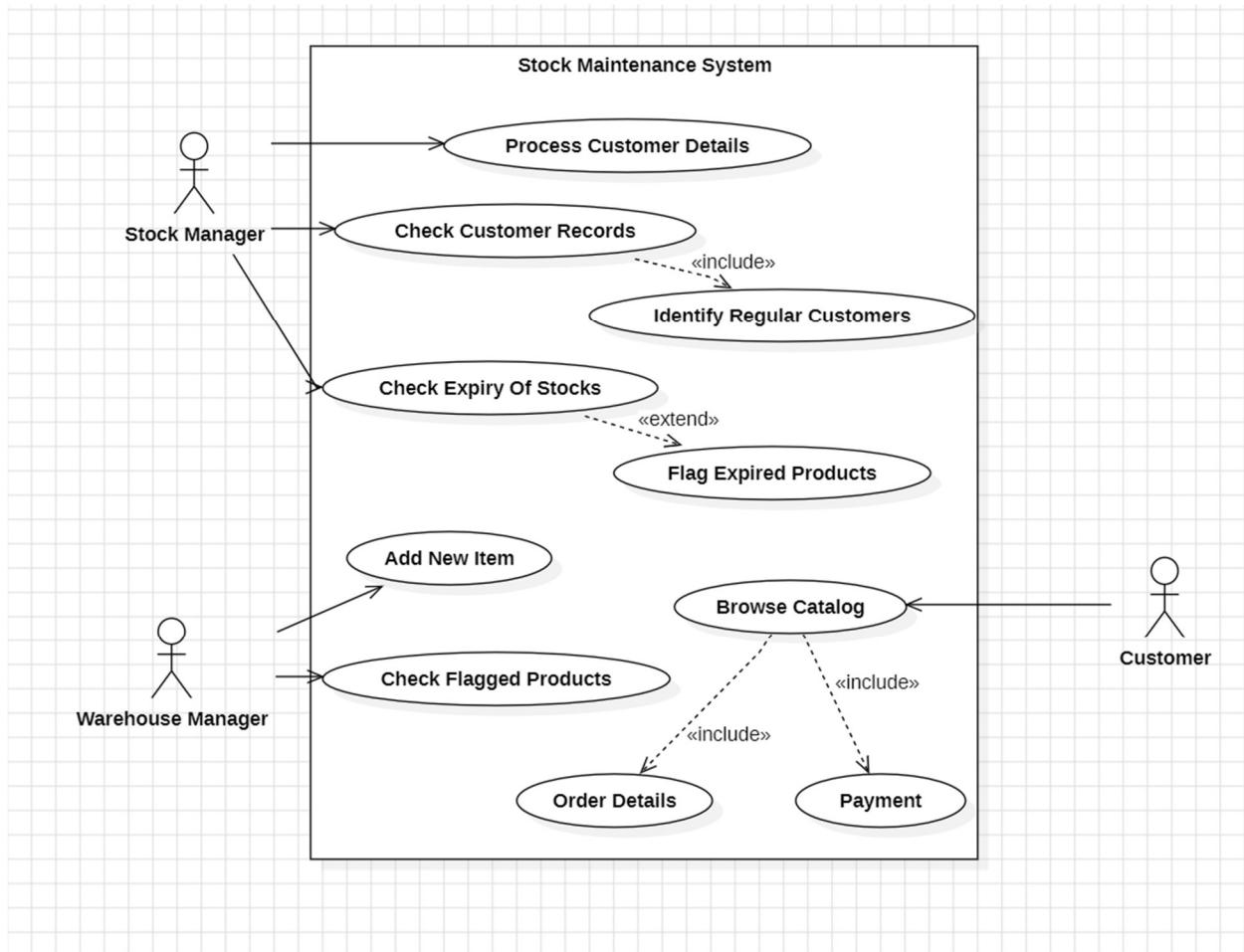


Fig 4.3 Stock Maintenance System - Use Case Diagram

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

Sequence Diagram

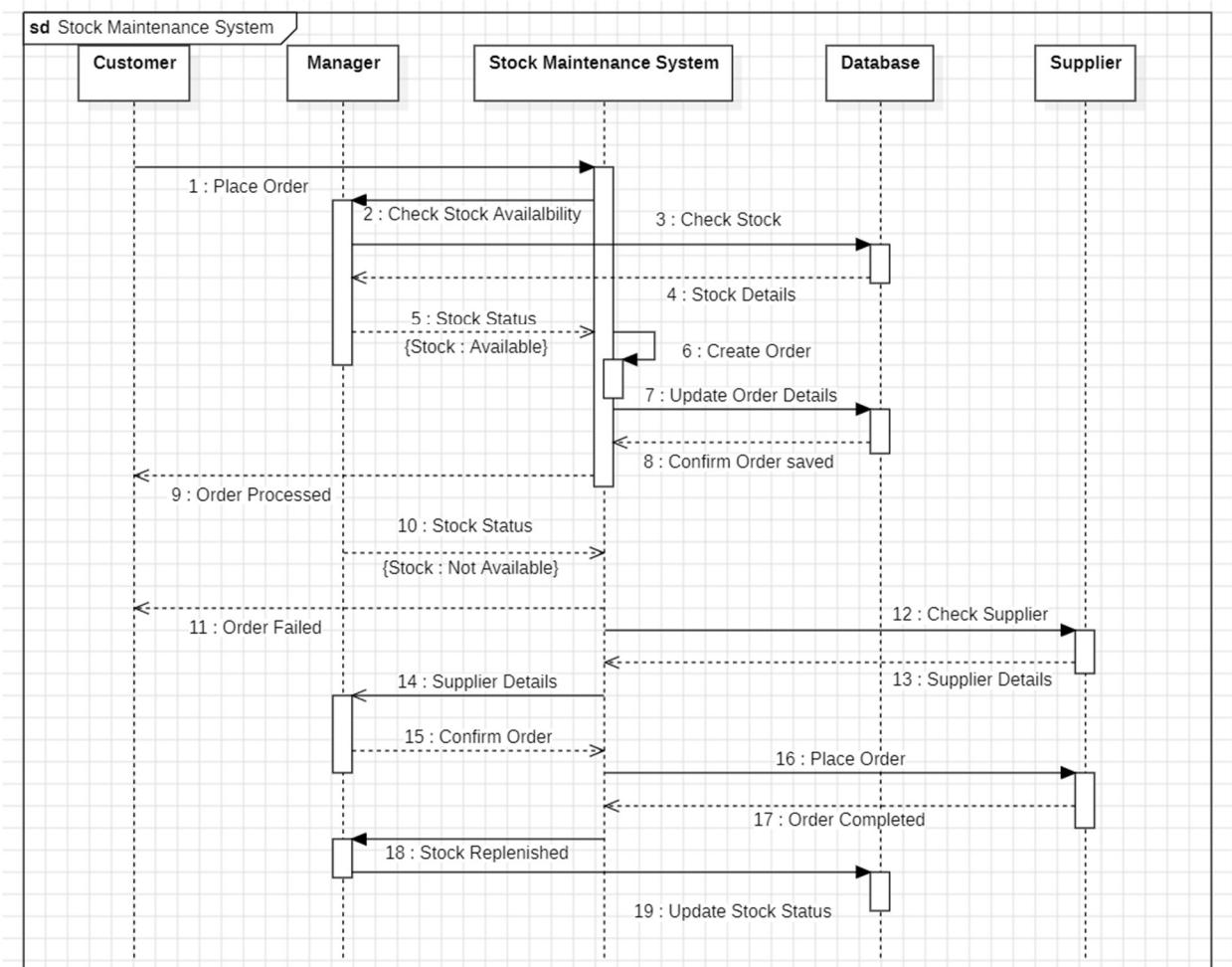


Fig 4.4 Stock Maintenance System - Sequence Diagram

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

Activity Diagram

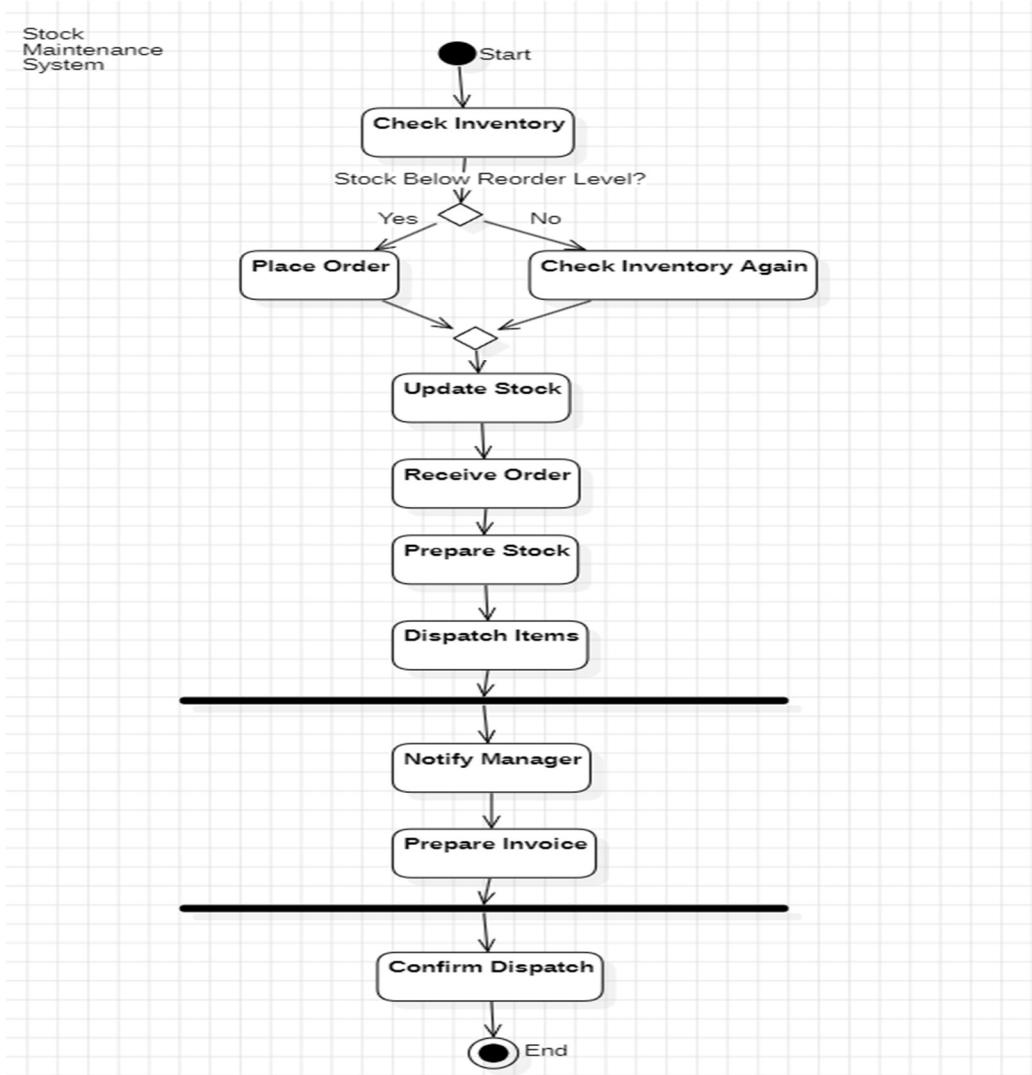


Fig 4.5 Stock Maintenance System-Activity Diagram

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

5. Passport Automation System

Software Requirement Specification

Date / /	
IV	Passport automation System
1.1	Introduction
1.1.1	Purpose
	Purpose of this document is to define functional and non-functional requirements of a passport automation system. It aims to streamline passport application processing and issuance by automating the workflow.
1.1.2	Scope
	The document provides the details for building automation system for end-to-end process of passport application, verification and issuance.
1.1.3	Assumptions
	The passport automation system allows citizens citizens to apply for passport online, track their status and schedule appointment for document verification.
2	General description
	The passport automation system will improve efficiency of application process by providing an online platform for applicants and officials.

Date _____

It will allow the applicants to submit requests, schedule appointments for document verification and track the status of their application.

3 Functional Requirements

- User can submit applications with required documents.
- User may schedule appointments for document verification.
- User can view the status of their application.
- User can get their application approved by officials & generate reports.

4 Interface requirements

- User interface should be web based for applicants to submit and track application.
- The government officials should be able to manage the applications.

5 Performance requirements

- Response Time: System should process user requests within a little time.
- System should maintain 99.9% uptime.
- System should handle around 500000 applications annually.

Date / /

6 Design Constraints

- System should comply with national process regarding issues of passport
- Data must be securely stored and maintained

V

7 Non functional attributes

- Security \Rightarrow All data transaction should maintain utmost security
- Reliability \Rightarrow System should handle user and official requests well
- Scalability \Rightarrow System should scale to meet growing needs.

8 Preliminary schedule and budget

Estimated development time is about 7 months with about 120000 dollars being the approximate budget.

Class Diagram

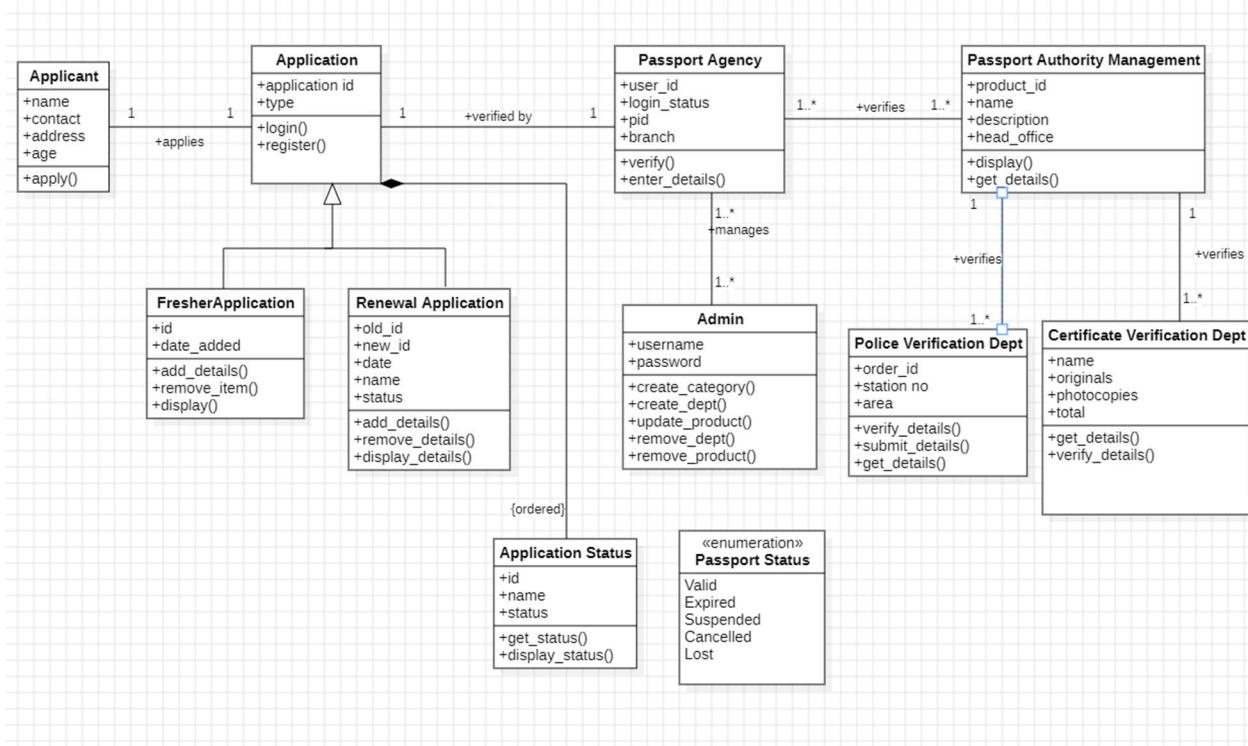


Fig 5.1 Passport Automation System - Class Diagram

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

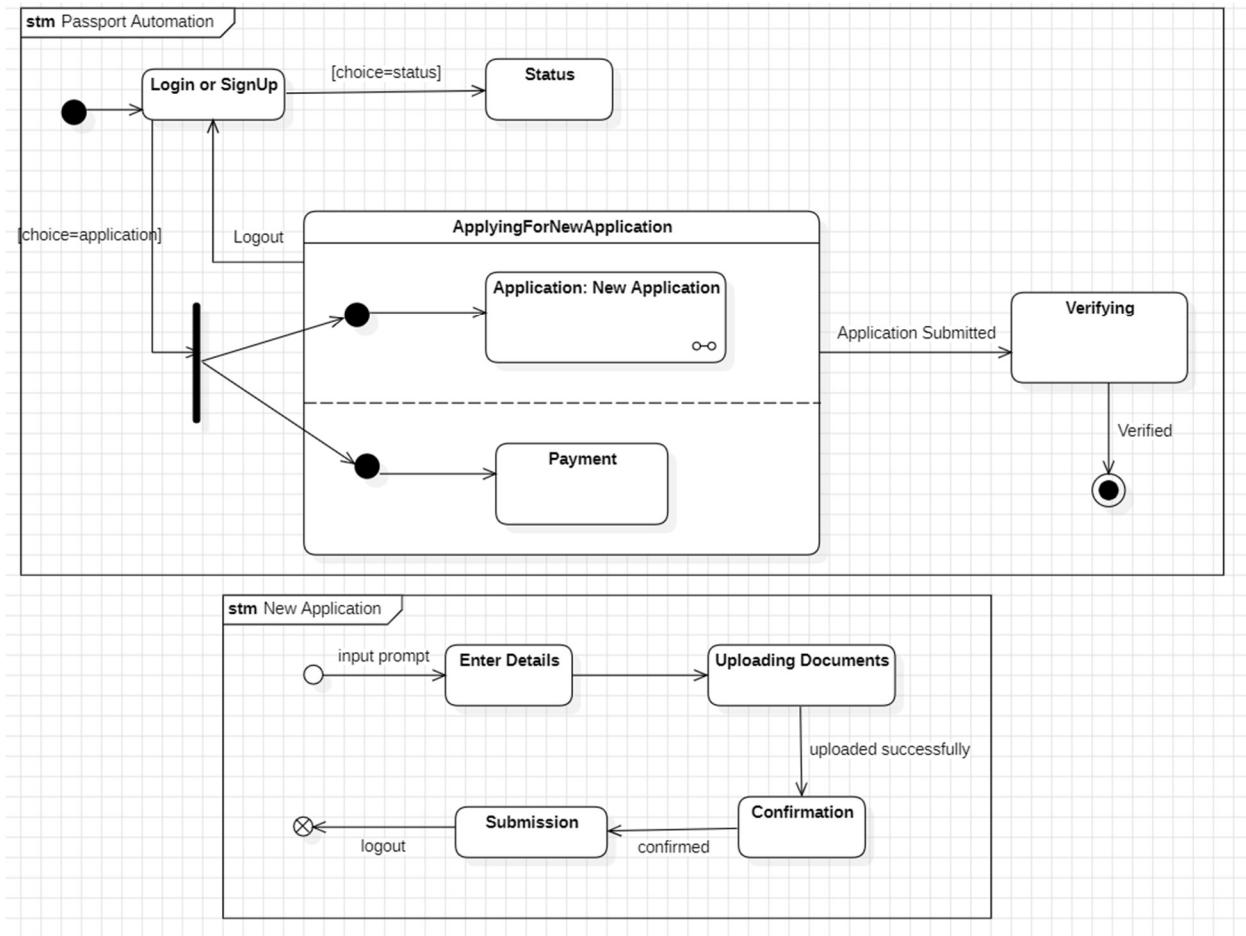


Fig 5.2 Passport Automation System - State Diagram

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

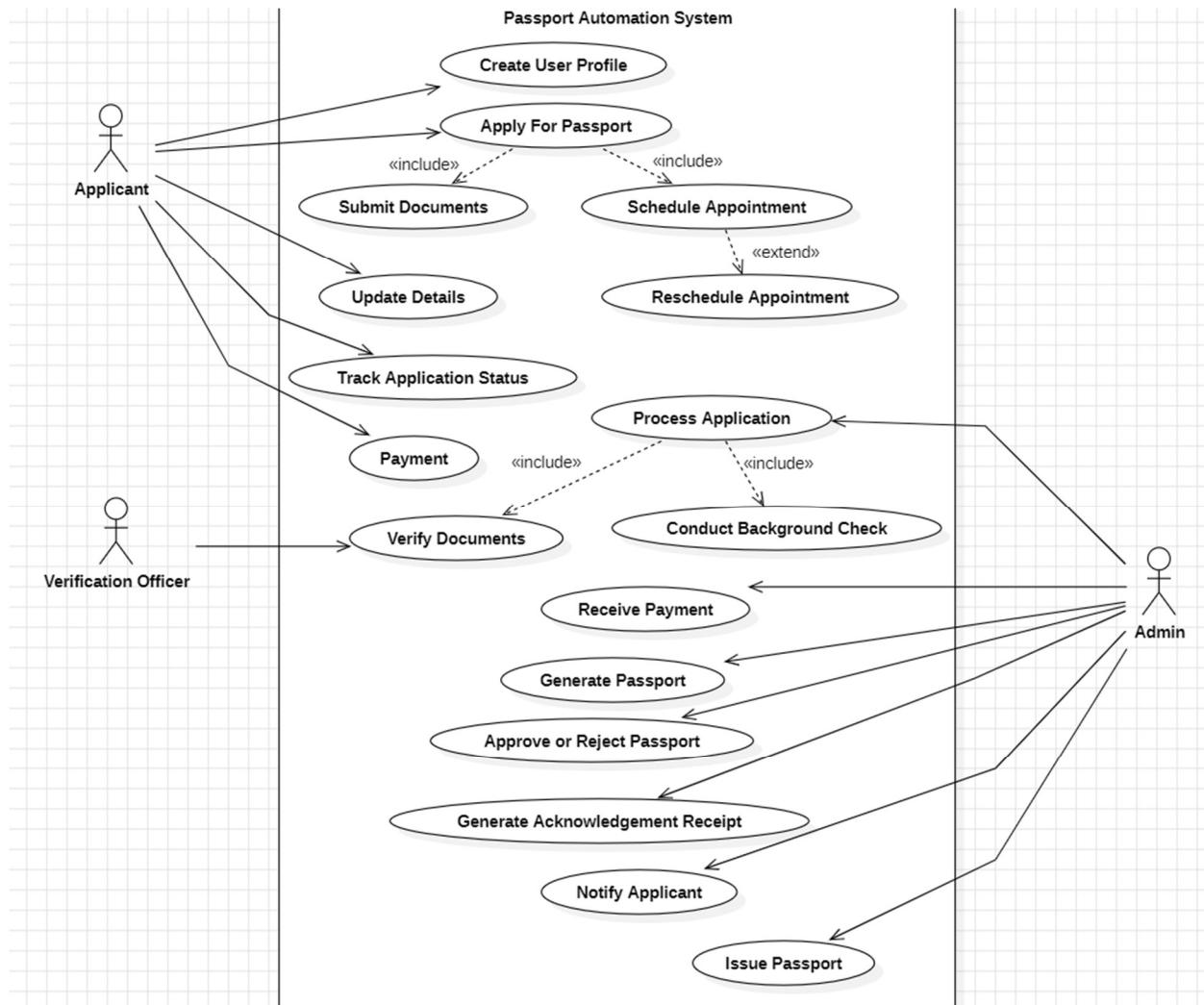


Fig 5.3 Passport Automation System - Use Case Diagram

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

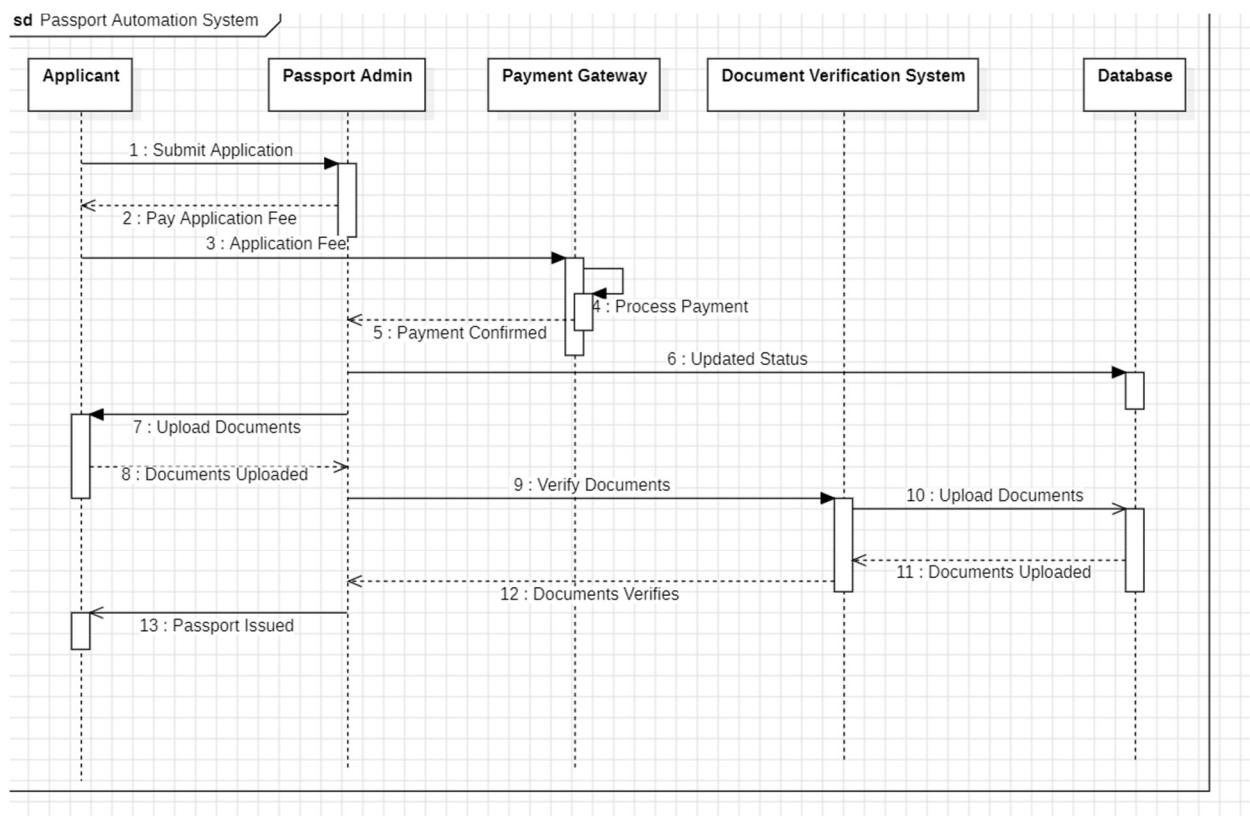


Fig 5.4 Passport Automation System - Sequence Diagram

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

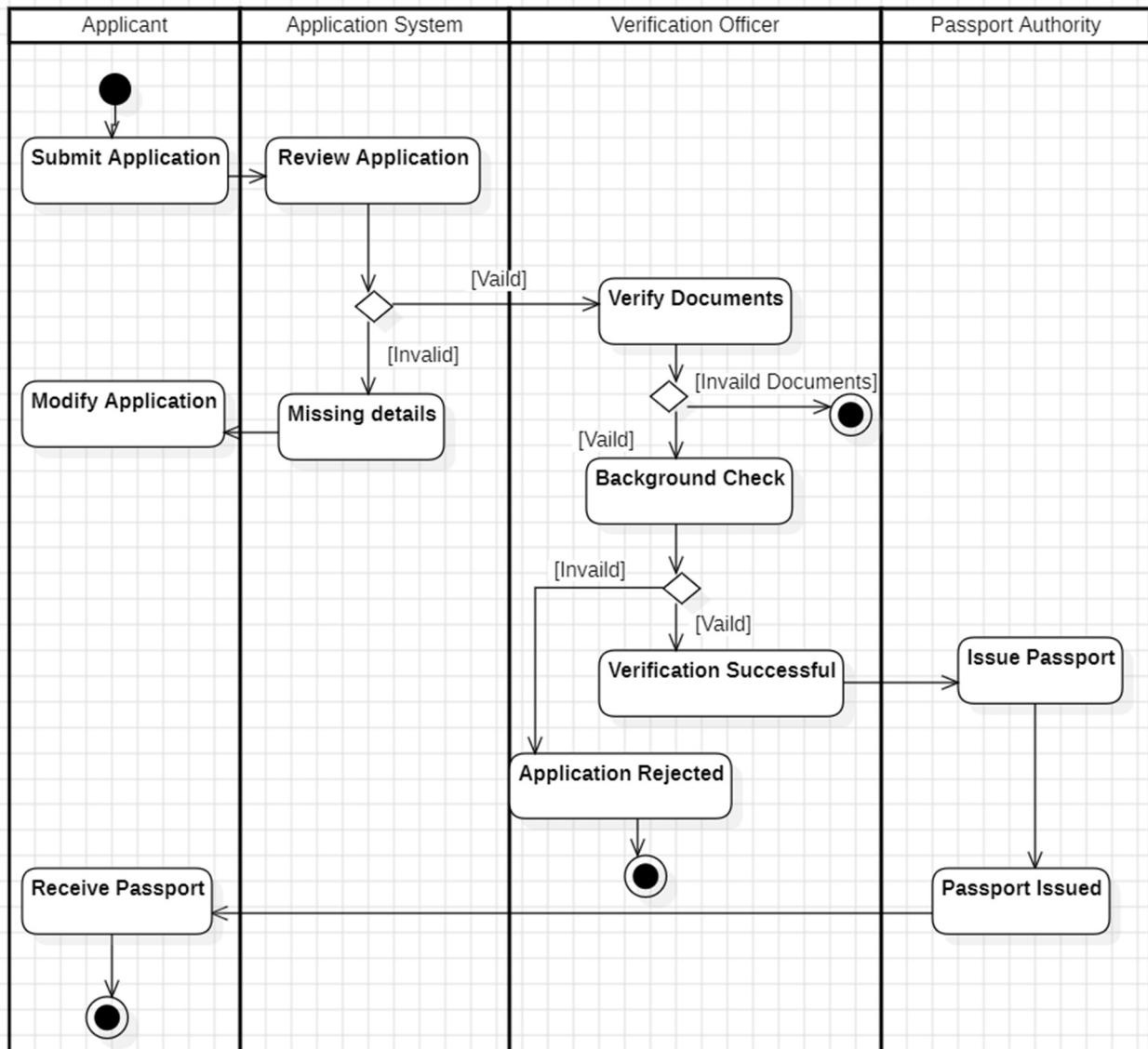


Fig 5.5 Passport Automation System - Activity Diagram

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.