

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.

Sohan A R

1BM22CS285

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

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



CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(22CS6PCSEO) laboratory has been carried out by **Sohan A R(1BM22CS285)** during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

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

Table of Contents

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

1. Hotel Management System

Software Requirement Specification

Date _____
Page _____

LAB - 2

HOTEL MANAGEMENT SYSTEM

1 INTRODUCTION

1.1 Purpose of this document:

The purpose of this document is to outline the software requirements for the ~~an~~ hotel management system. It serves as a comprehensive guide for stakeholders, including developers, project managers, and clients, to understand the functionalities, features, and constraints of the system.

1.2 Scope of the document:

The Hotel Management System (HMS) is designed to cater all the residents of the hotel as well as its staff by managing the finances, delegation of staff, ~~an~~ ~~an~~ hospitality management, background verification of residents, payment processing of tariffs of rooms etc.

1.3 Overview:

The ~~an~~ Hotel Management System (HMS) provides an one stop solution ^{user interface} by managing the payments, staff attendance, staff allotment, incentive schedule and over time payment, it also ~~an~~ manages the number of residents in the hotel rooms by acting as a bridge between the hotel staff and its residents.

2 General description:

The HMS aims to streamline the management protocols of the hotel industry. Its key objectives include:

User characteristics: ~~an~~ Individuals staying ~~an~~ in hotels for ~~an~~ vacation.

Hotel staff : Manage customers and maintain the hotel.

features -

User friendly interface to manage the staff and customers of the hotel. And ~~receives~~ maintenance of the hotel ~

3) Functional Requirements

- (i) User registration and Authentication
- (ii) Reservation management
- (iii) check-in | check-out process
- (iv) Billing management
- (v) Room Management .

4) Interface Requirements

- (i) User interface (UI) : Provide a user friendly interface for the users to interact with
- (ii) API's : The API's provide improved integration and scalability.
- (iii) Database interface : A proper database interface allows the hotel management to feed the related statistics onto the database

5) Performance requirements

- (i) Response time
- (ii) Concurrent users
- (iii) Data storage

6) Design constraints

- (i) Platform
- (ii) Database
- (iii) Framework

7) Non-functional attributes

- (i) Security

- ii) Scalability
- iii) Reliability

8) Preliminary Schedule and Budget:
i) Estimated Development duration

9) ~~Estimated cost~~

8) Preliminary ~~cost~~ Schedule and Budget

i) Estimated development duration : 1 month

ii) Estimated Budget : \$340,000

iii) Testing and deployment : \$100,000

iv) ~~estimated~~ Project management : \$150,000

new date

v) Development : \$90,000

Q
2/10/20

Class Diagram

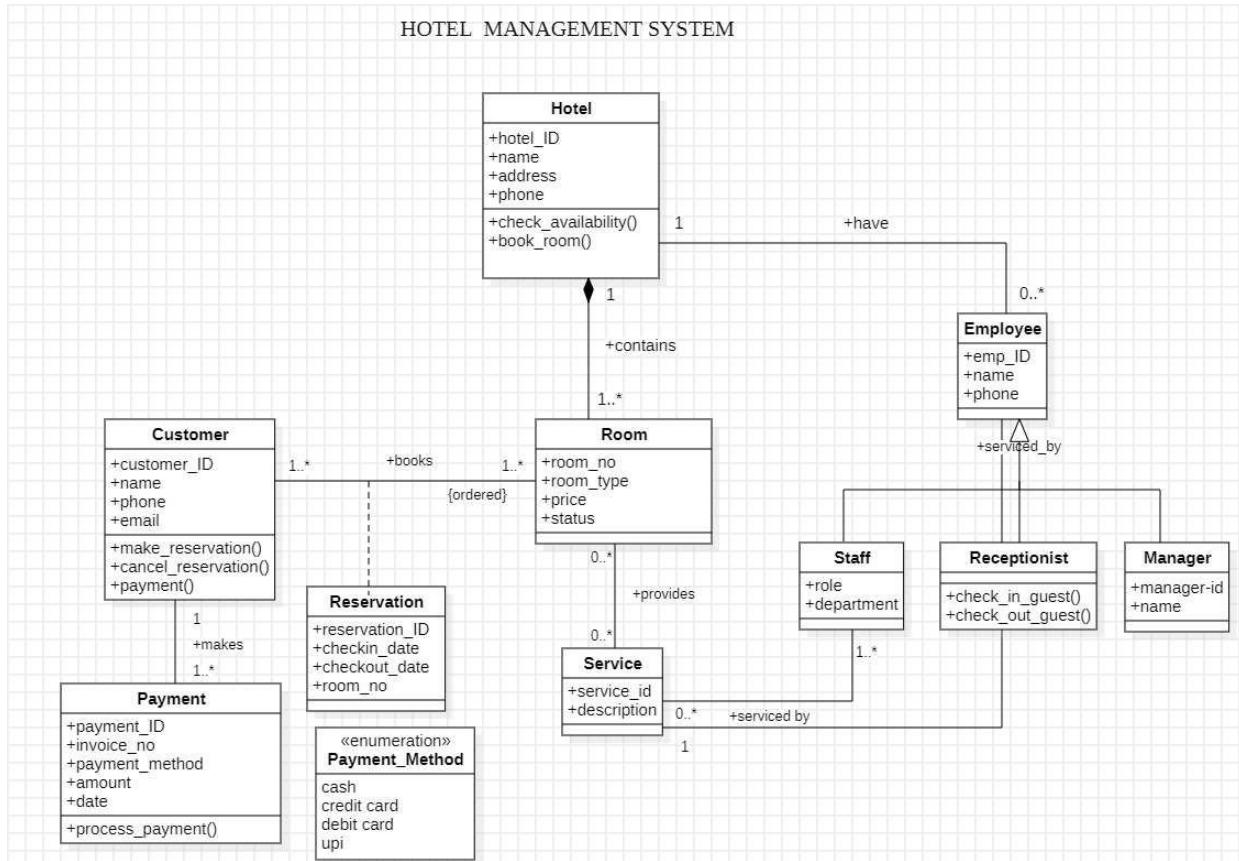


Fig1.1 Hotel Management System - Class Diagram

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

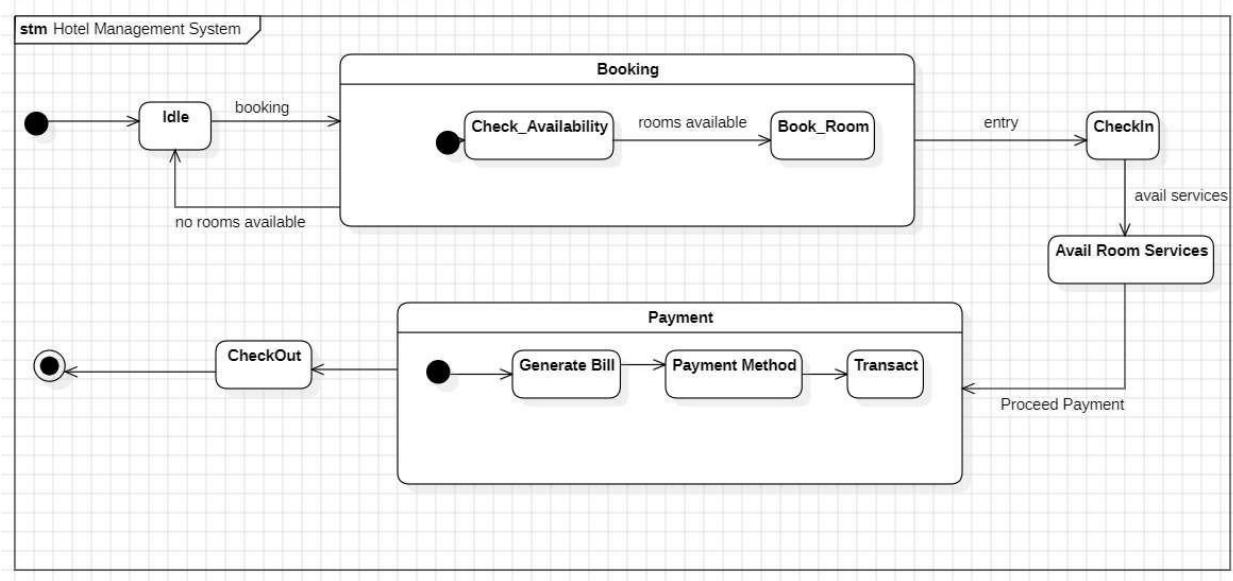


Fig1.2 Hotel Management System - State Diagram

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

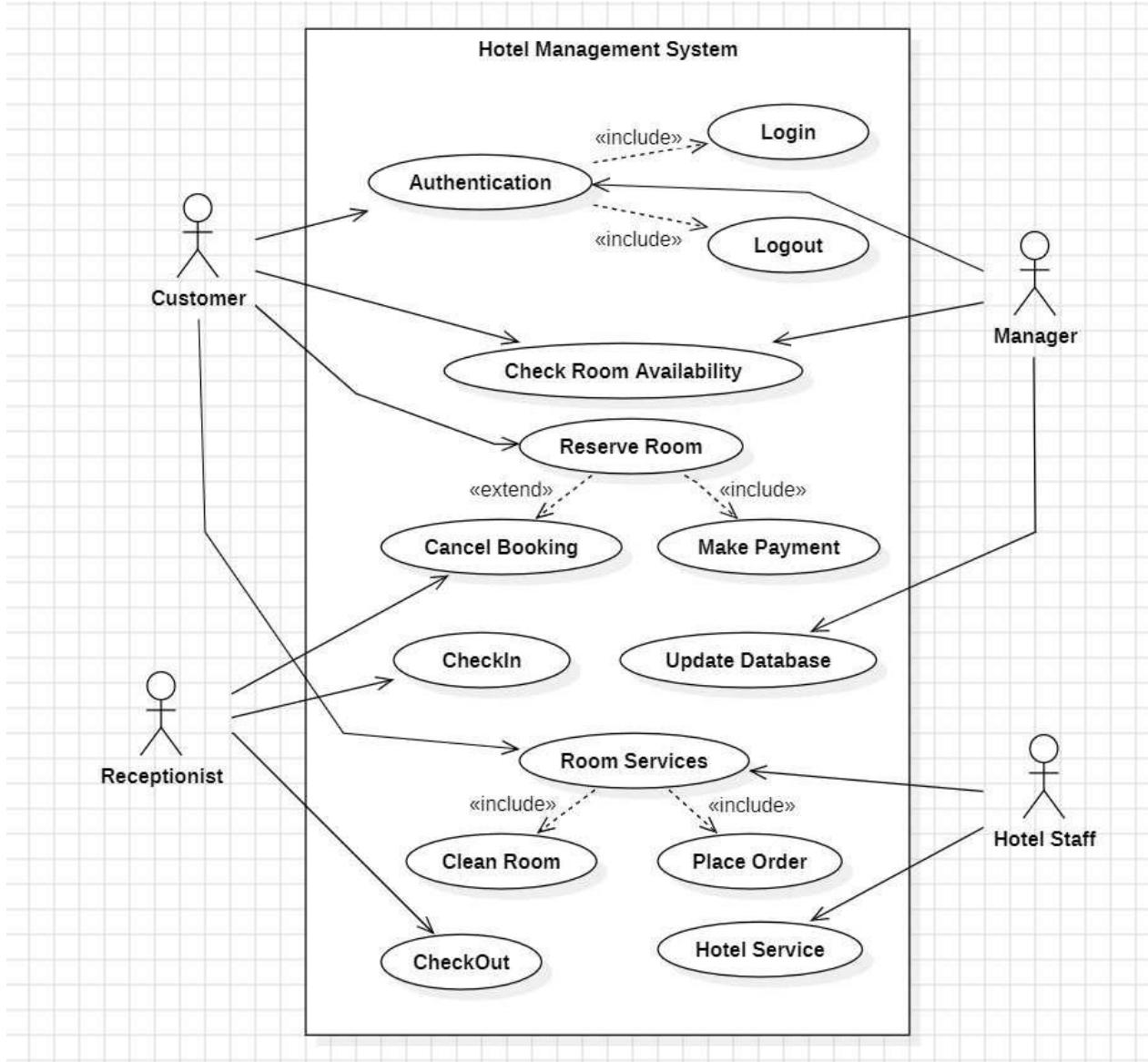


Fig1.3 Hotel Management System - Use Case Diagram

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

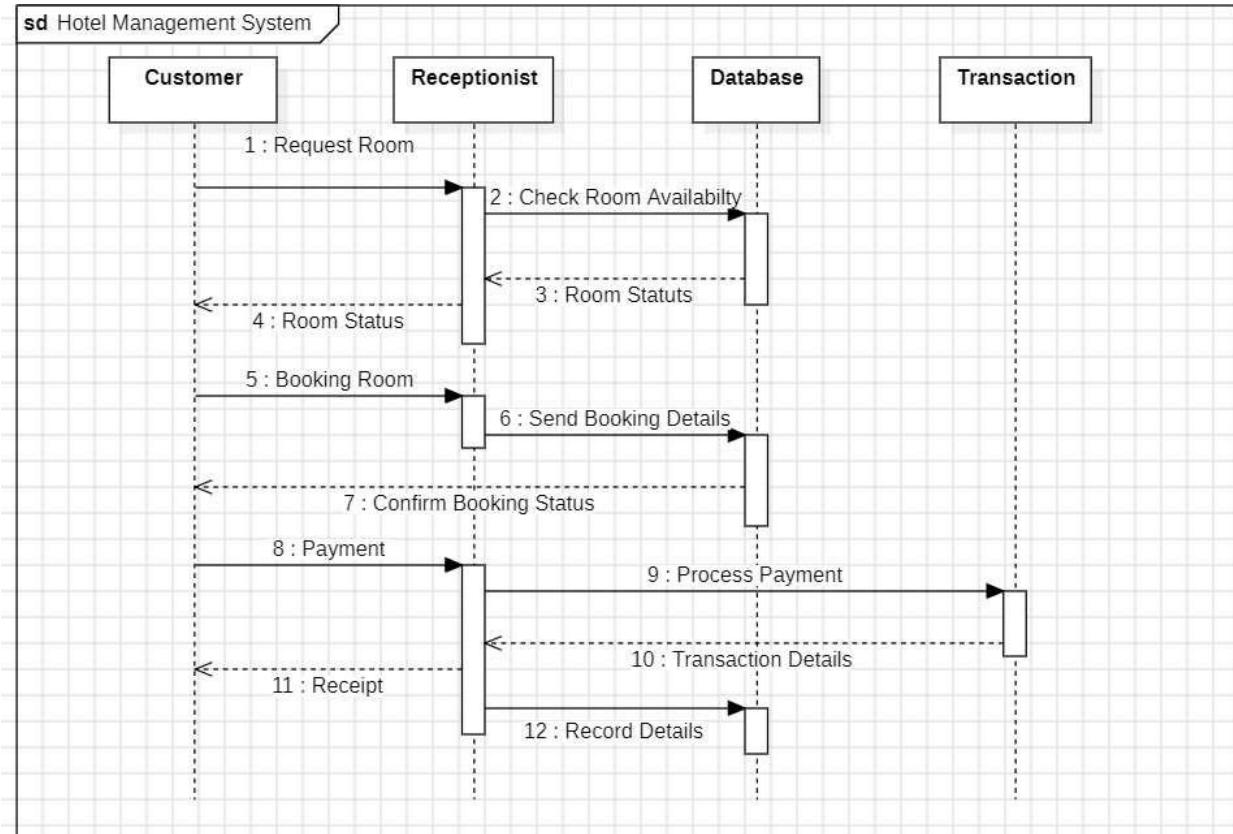


Fig1.4 Hotel Management System - Sequence Diagram

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

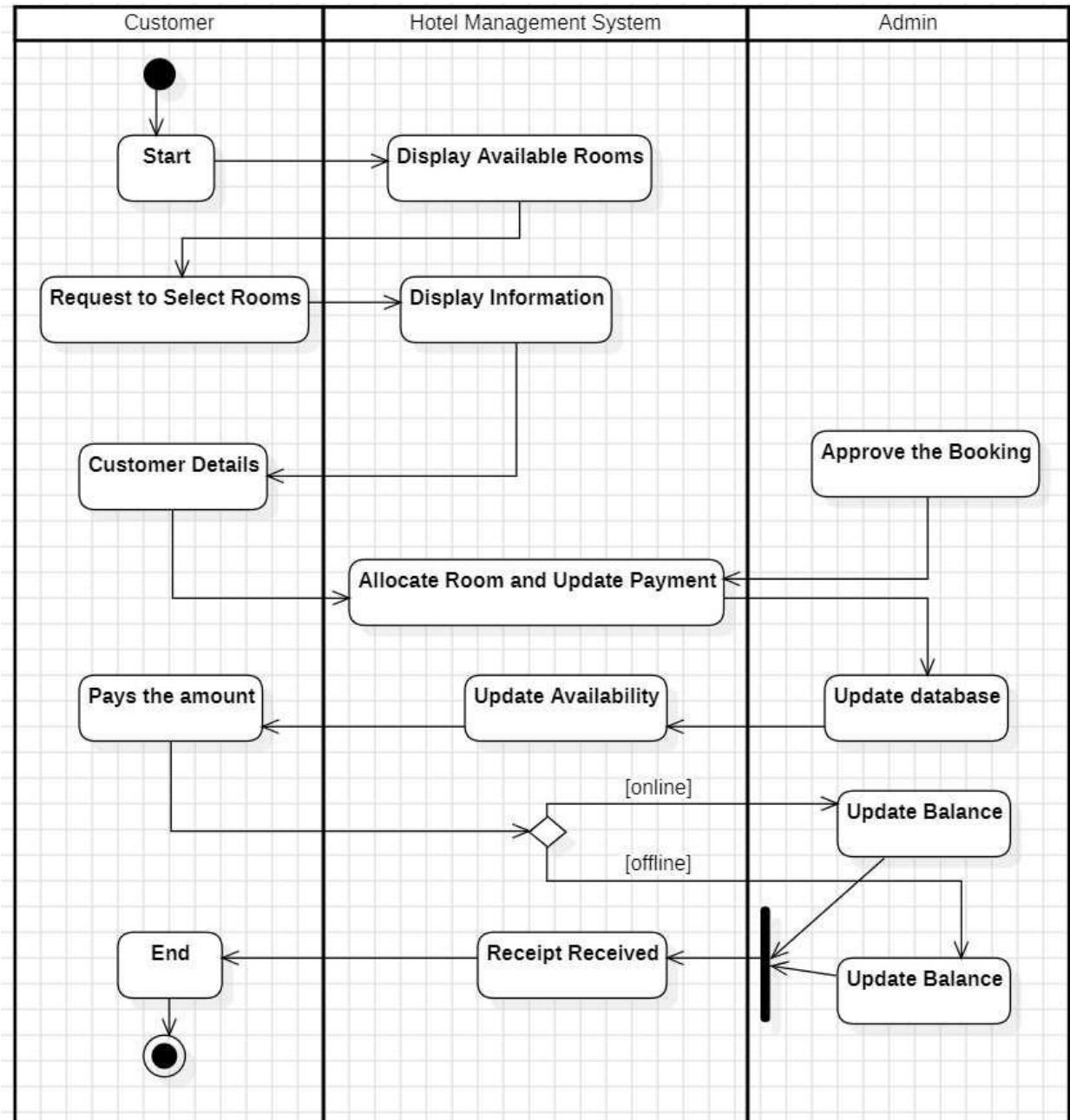


Fig1.5 Hotel Management System - Activity Diagram

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

CLASSMATE
Date _____
Page _____

20/9

Software Requirement Specifications (SRS)

1. Introduction

2. Purpose of this document

The purpose of this document is to outline the software requirements for the ~~Project Credit card processing system~~. It serves as a comprehensive guide for stakeholders, including developers, project managers, and clients, to understand the functionalities, features, and constraints of the system. The document will ensure that all parties have a clear understanding of the project objectives and deliverables.

3. Scope of this Document

The CCPS is designed to facilitate secure and efficient credit card transactions for businesses. It encompasses features such as transaction processing, fraud detection, reporting, and integration with various payment gateways. The estimated development cost is \$200,000, with a timeline of 8 months of completion.

4. Overview

The credit card processing system will enable businesses to accept credit card payments seamlessly. It will provide a secure interface for transaction processing, ensure compliance with financial regulations, and include features for reporting and fraud detection. The system will enhance user experience.

5. General Description:

The CCPS aims to streamline the credit card payment process for businesses. Key objectives include:

6. User characteristics:

~~1. Merchants~~ User characteristics:
Merchants: Business owners and staff who will process transactions

Customer: Individuals making purchases using credit cards.

Features:

User friendly interface for processing payments.
Support for various credit card brands and payment methods.

Benefits:

Increased sales through flexible payment options
Enhanced security and fraud prevention

Importance:

Providing a reliable credit card processing solution is critical for businesses to thrive in a competitive market.

3. Functional Requirements:

User Authentication: Secure login for merchants and administrators.

Transaction processing: Ability to process credit card transactions in real time.

Request management: Facilitate processing of requests and chargebacks.

Fraud detection: Implement mechanisms to detect and prevent fraudulent transactions.

Reporting: Generate transaction reports, including sales summaries and dispute resolutions.

Integration: API for integration with e-commerce platforms and point of sale (POS) systems.

4. Interface Requirements

The CCRS will include the following interfaces

Merchant Interface:

A web-based dashboard for merchants to manage transactions and access reports.

Customer Interface : A secure payment portal for customers to enter credit card details.

API Interface : RESTful API's for third party ~~operators~~ integrations and payment gateway connections.

5 Performance Requirements

The system should meet the following performance criteria:

Response time : All transaction requests should be processed within 3 seconds.

Concurrent Users : Support up to 500 concurrent transactions without performance issue.

6 Design Constraints

The design of the CCPS will adhere:

Technology Stack : Must utilize specified technologies
Compliance : Must adhere to payment card industry data security standards

7 Non Functional Attributes

Security : Implement system encryption protocols

Usability : Intuitive interface for ease of use by merchants

Portability : Compatible across multiple platforms

8 Preliminary Schedule and Budget

The project is scheduled for a duration of 8 months, with following milestones:

Requirements Gathering : 1 month

Design Phase : 2 months

Development phase: 4 months

Testing and Deployment: 1 month

Budget Estimate

Total Estimated Cost: \$200,000

Development: \$120,000

Testing: \$40,000

Project Management: \$40,000

✓
3/09

Class Diagram

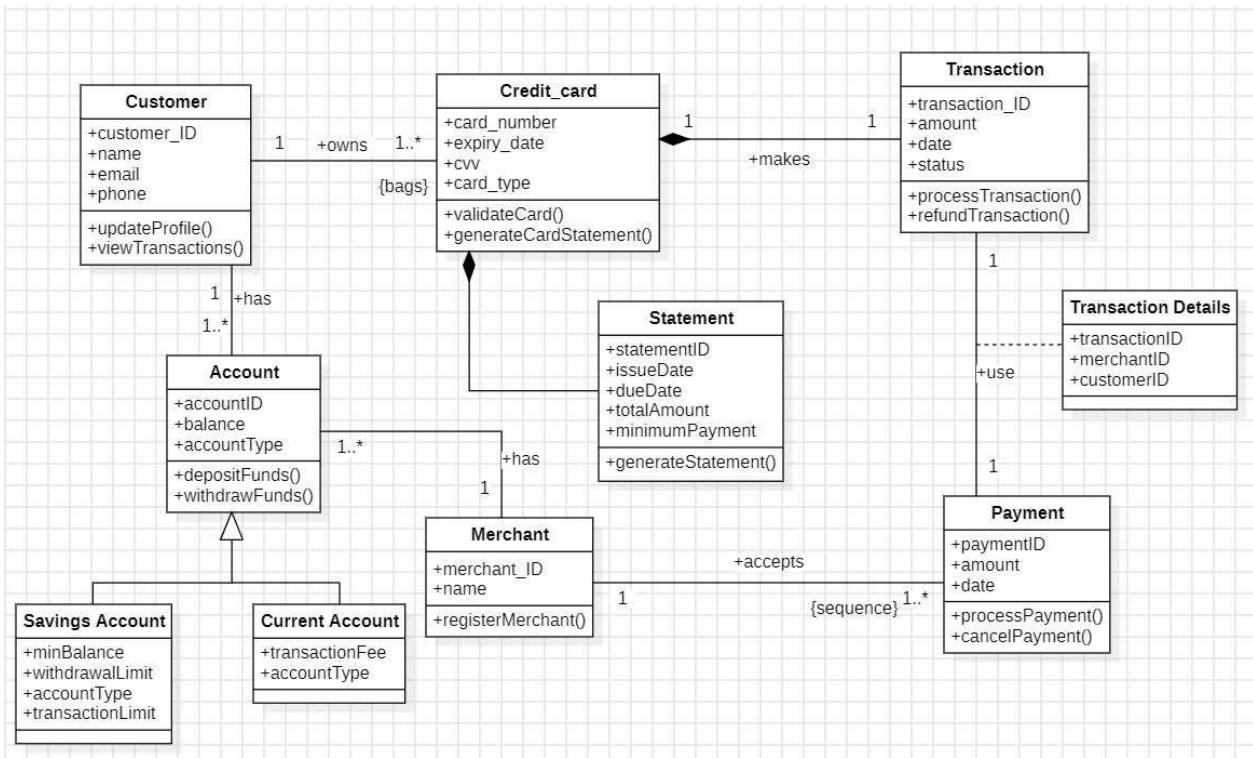


Fig 2.1 Credit Card Processing System - Class Diagram

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

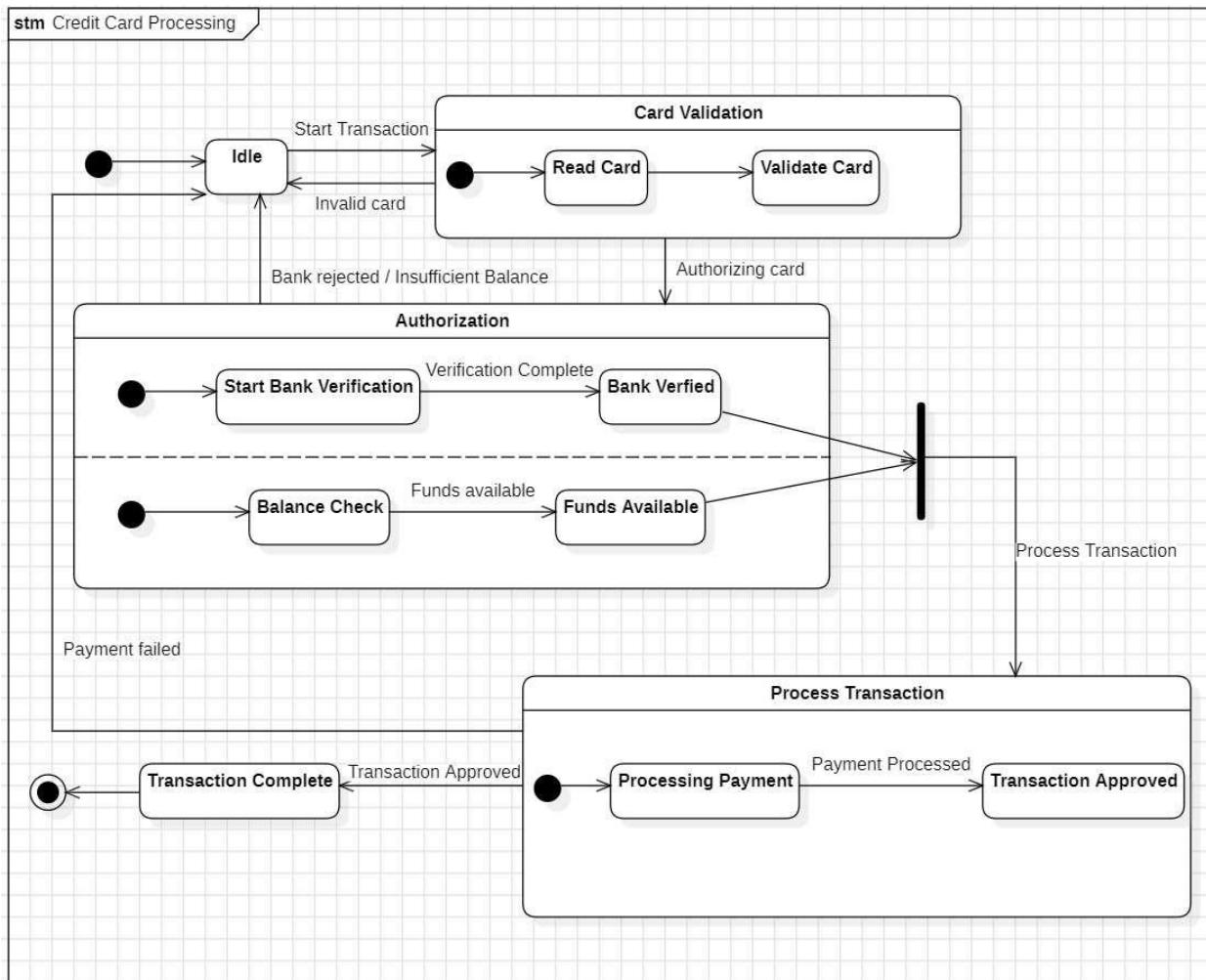


Fig 2.1 Credit Card Processing System - State Diagram

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

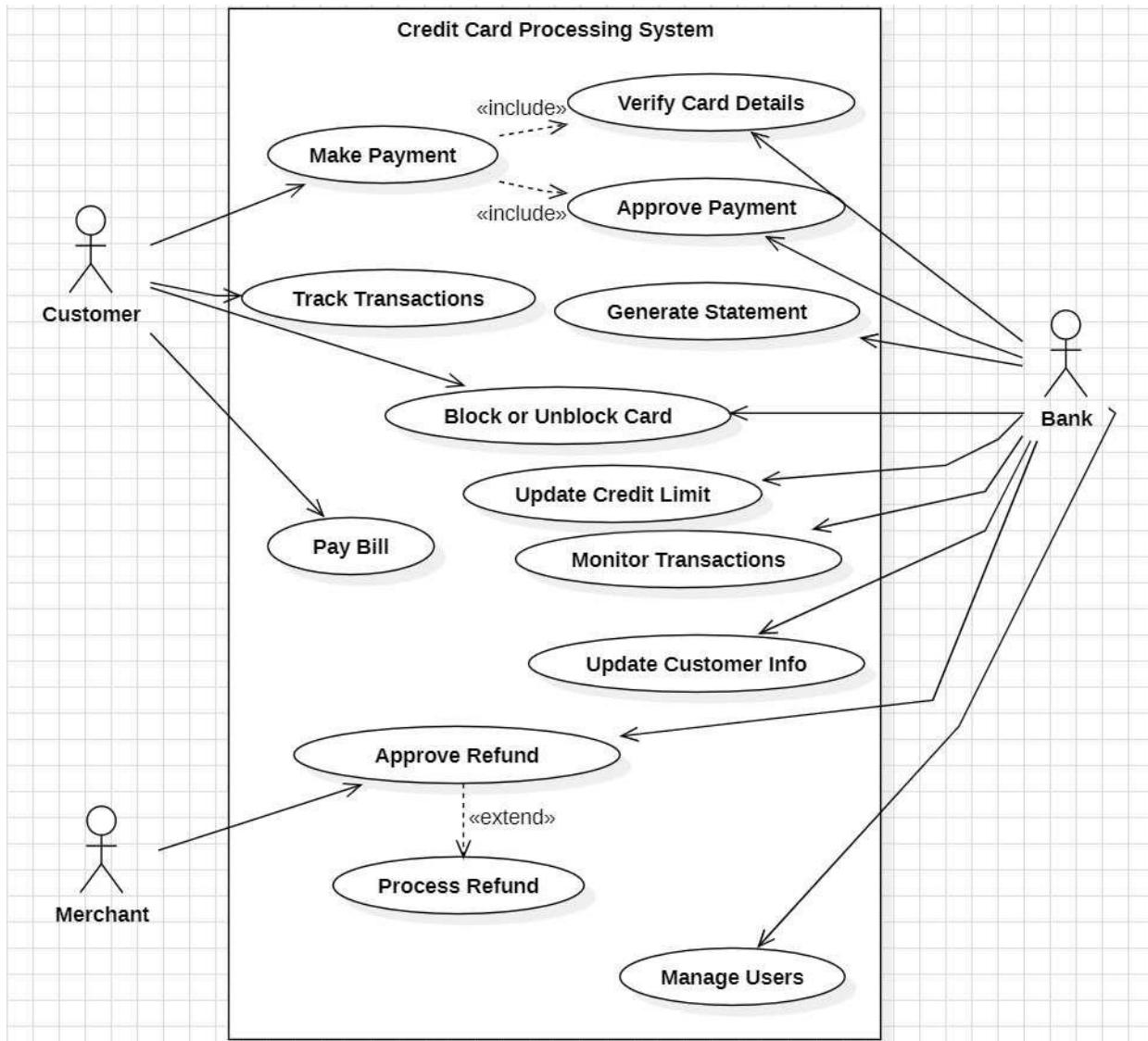


Fig 2.1 Credit Card Processing System - Use Case Diagram

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

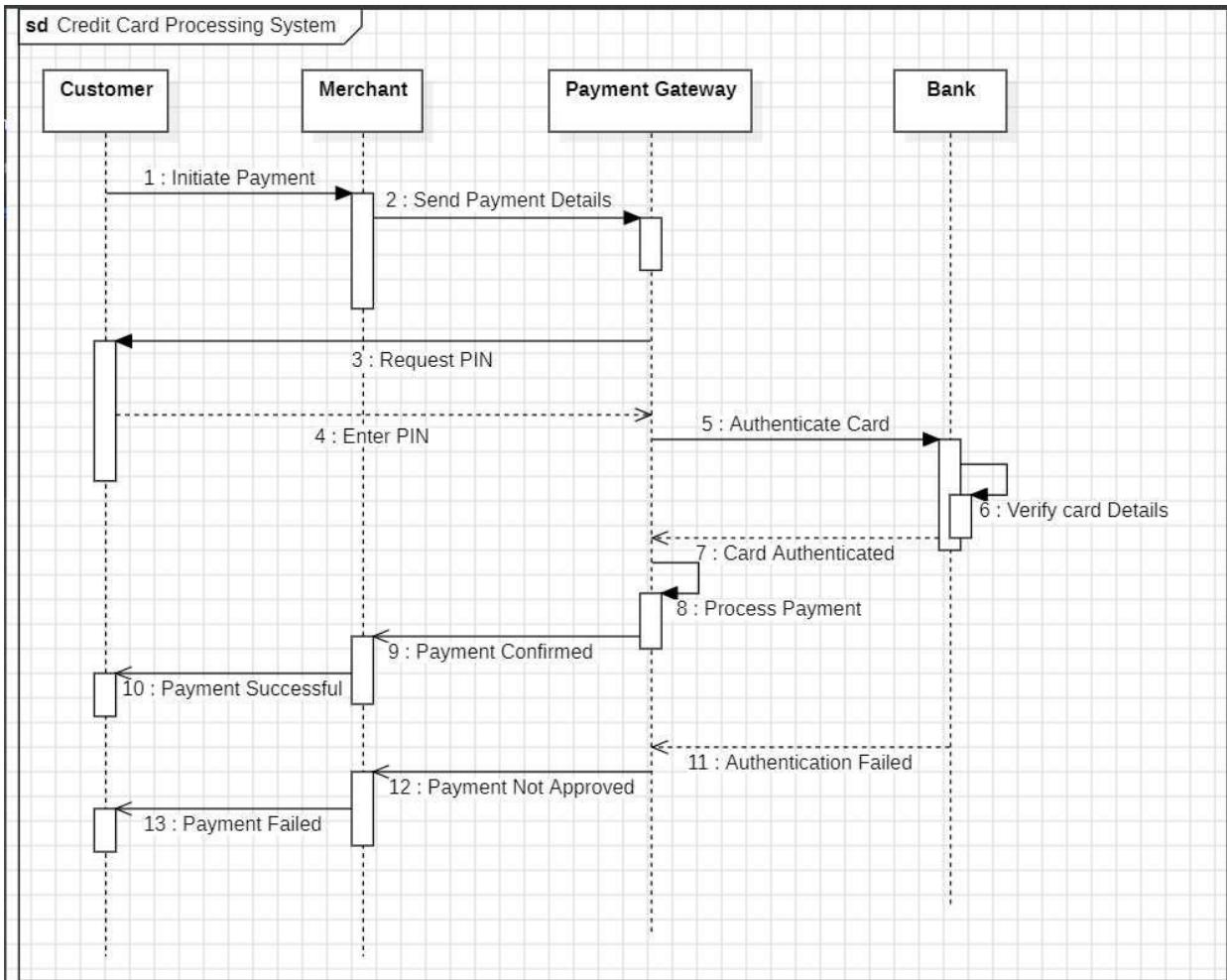


Fig 2.1 Credit Card Processing System - Sequence Diagram

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

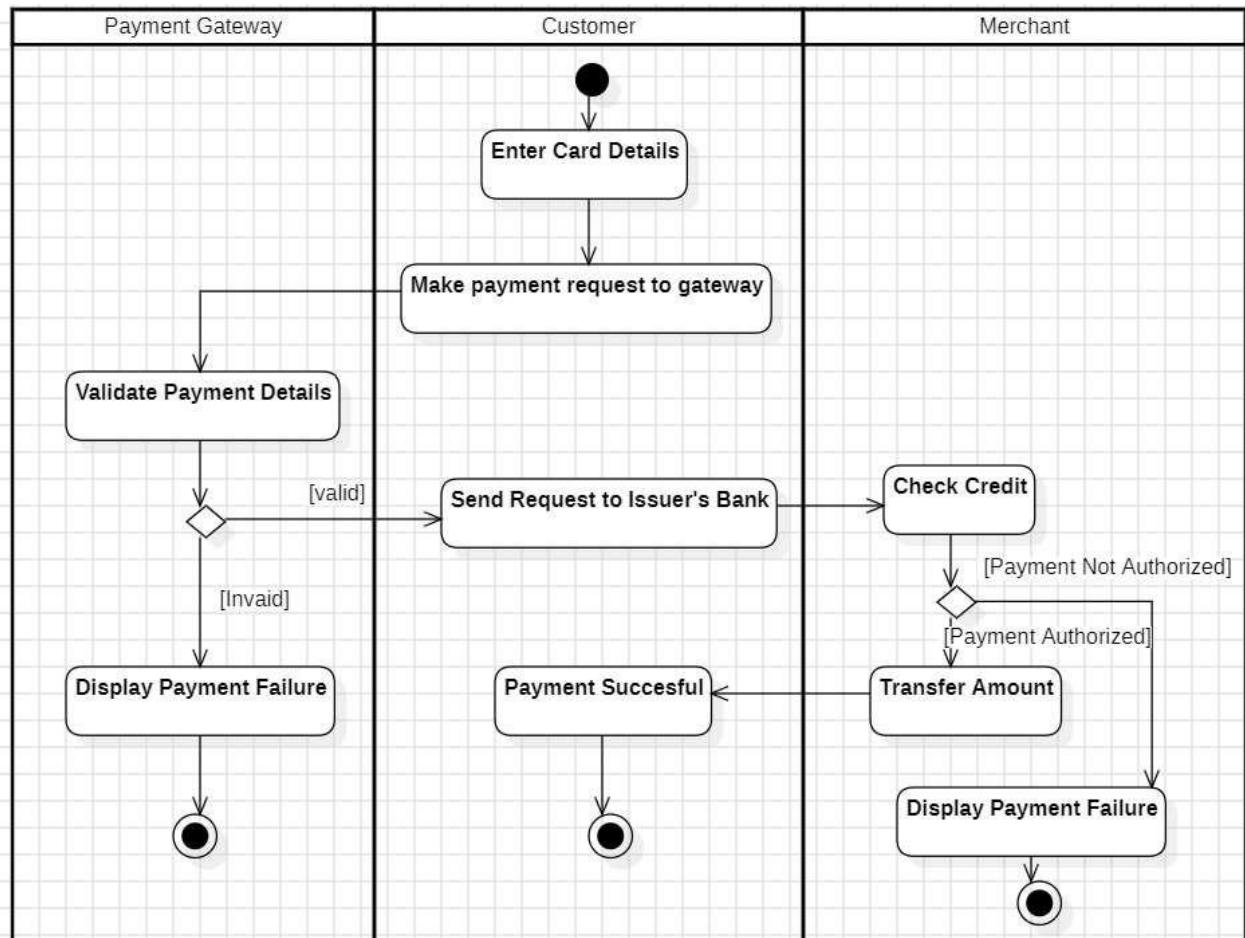


Fig 2.5 Credit Card Processing System - Activity Diagram

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

Software Requirement Specification (SRS) LIBRARY MANAGEMENT SYSTEM	
1.1	1. Introduction
1.1.1	Purpose of this document : The purpose of this document is to outline the requirements for the library management system. It serves as a comprehensive guide for its stakeholders, including developers, project managers, and clients, to understand the functionalities, features, and constraints of the system.
1.1.2	Scope of this document : The Library Management System (LMS) is designed to manage and automate the library resources, including cataloging, user management, and transaction processing. This document details the project objectives, expected benefits, estimated costs and timelines, ensuring value for the library and its users.
1.1.3	Overview : The library management system will serve as a centralised platform for managing library operations, enhancing efficiency, improving user experience, and providing tools for library staff to manage resources effectively.
2	General Description
	The LMS targets library staff and members, providing features such as book cataloging, membership management, lending and returning books, and generating reports. The system's primary objective is to simplify library operation, improve resource accessibility.
3	Functional Requirements
3.1	User registration and Authentication

- (i) Book Cataloguing
- (ii) Memberships management
- (iii) Books Lending

4.) Interface Requirements

- i) User Interface (UI)
- ii) API's
- iii) Database Interface

5.) Performance Requirements

- (i) Response Time
- (ii) Concurrent Users
- (iii) Data storage

6.) Design Constraints:

- (i) Platforms
- (ii) Database
- (iii) Framework

7.) Non functional Attributes

- i) Security
- ii) Reliability
- iii) Scalability

8.) Preliminary Schedule and Budget

- i) Estimated Development duration : 2 months
- ii) Estimated Budget : \$200,000
- iii) Testing and Deployment : \$100,000
- iv) Project management : \$50,000
- v) Development : \$150,000

Class Diagram

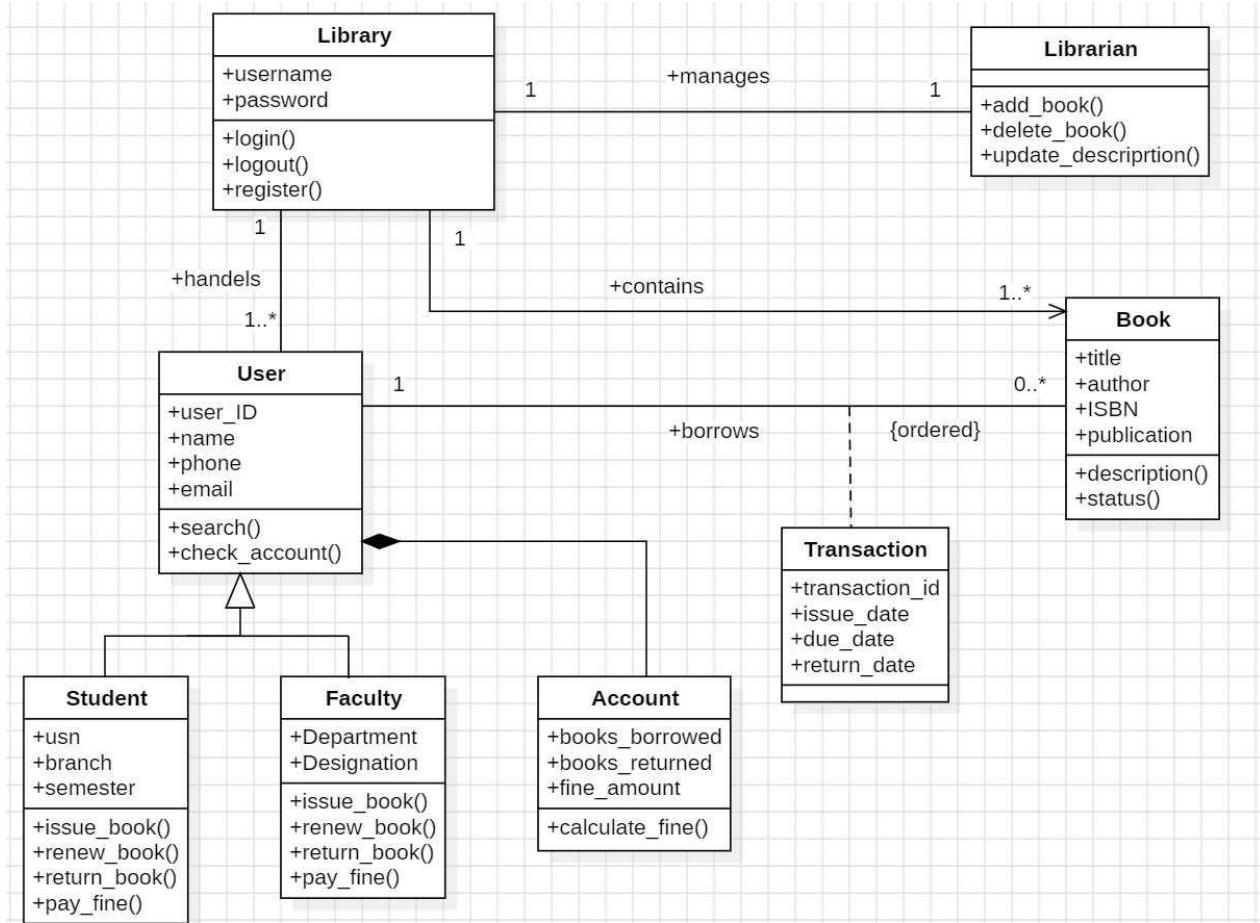


Fig 3.1 Library Management System -Class Diagram

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

State Diagram

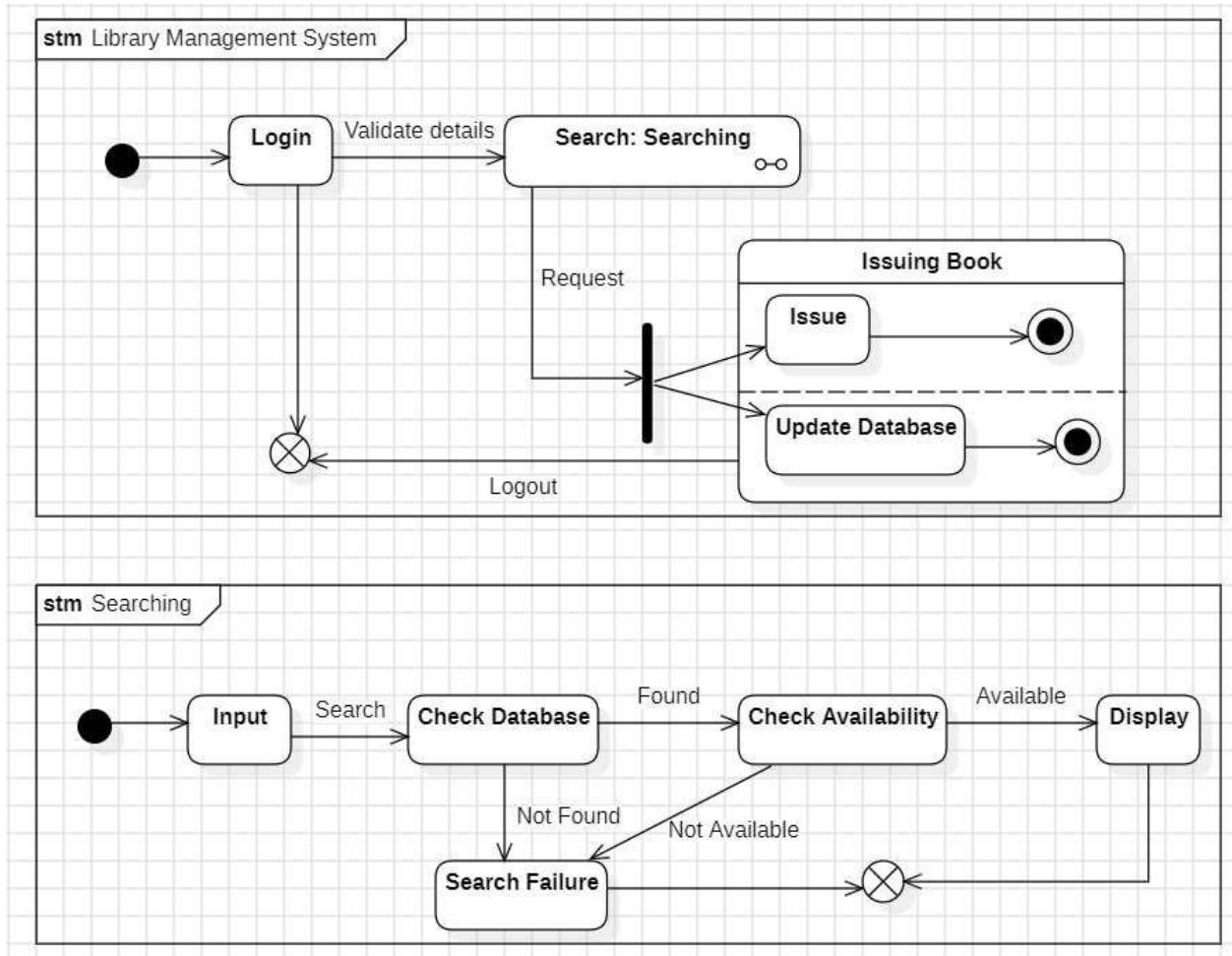


Fig 3.2 Library Management System - State Diagram

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

Use Case Diagram

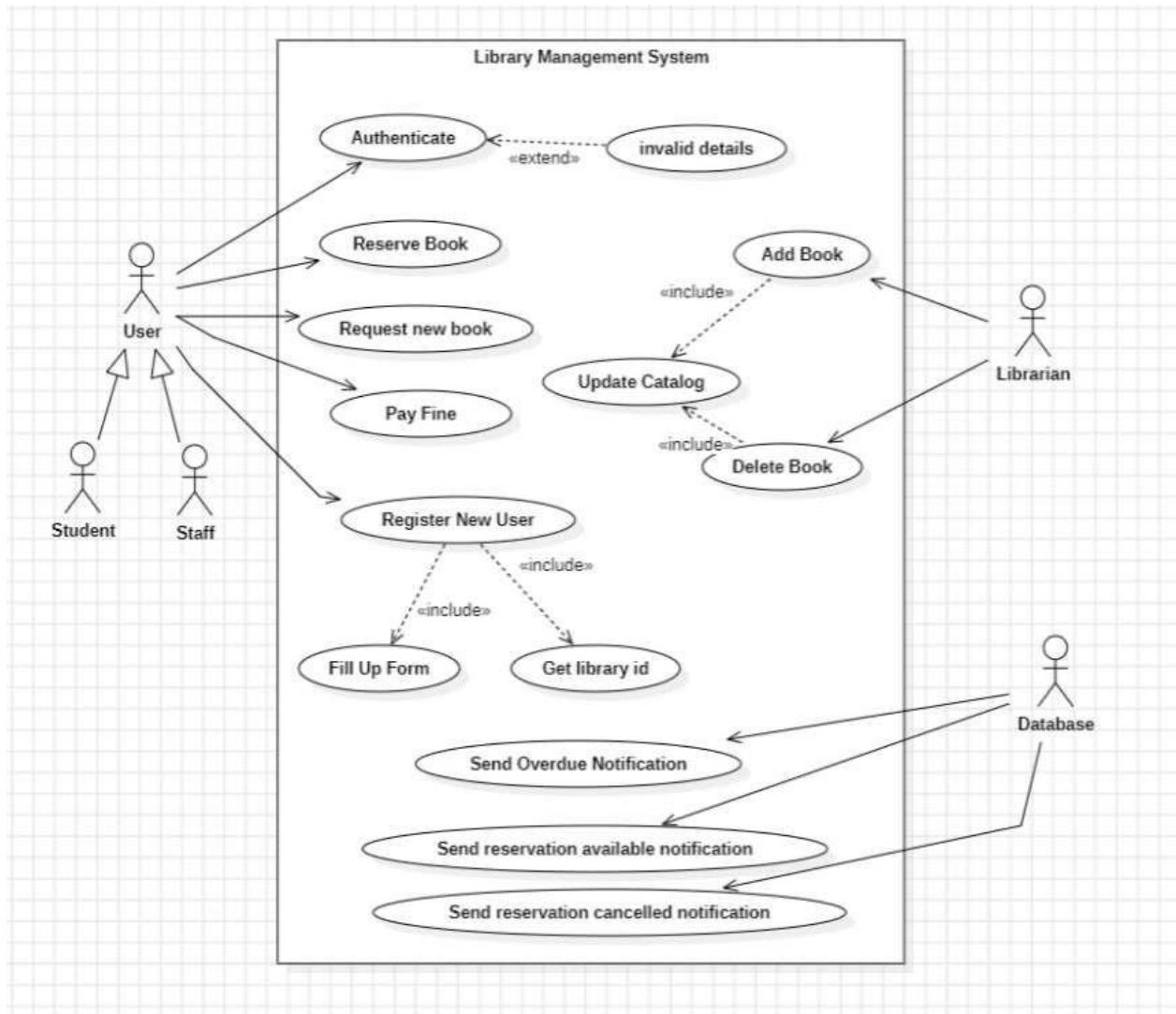


Fig 3.3 Library Management System - Use Case Diagram

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

Sequence Diagram

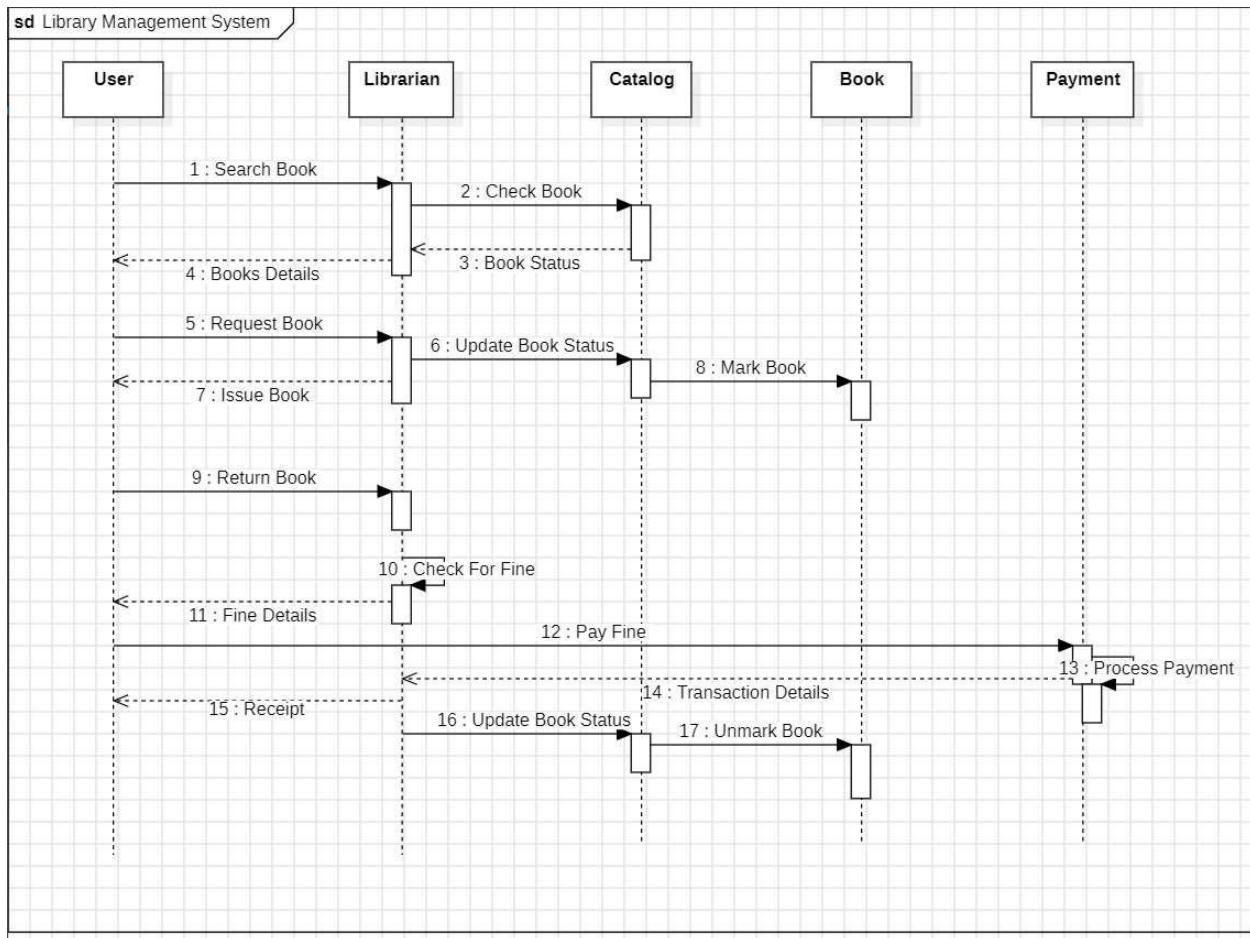


Fig 3.4 Library Management System - Sequence Diagram

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

Activity Diagram

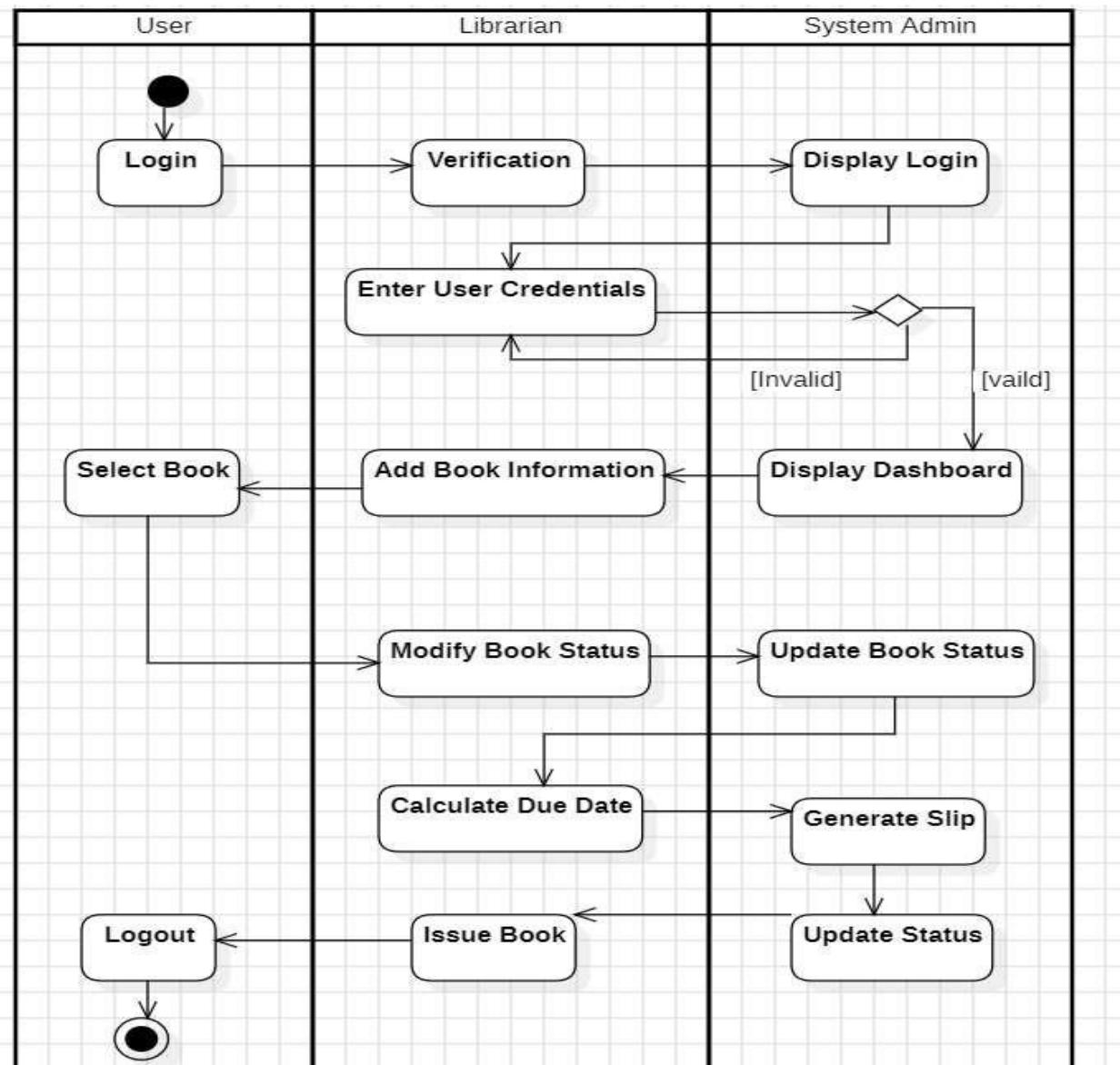


Fig 3.5 Library Management System - Activity Diagram

The activity diagram outlines the workflow of a library management system. It starts with a user logging in, followed by credential verification. Successful login grants access to book selection for the user. Simultaneously, librarians can add new books or modify existing book information. System administrators possess the authority to update book statuses and generate slips related to book transactions. Upon book selection, the system calculates the due date and issues the book to the user, updating the database accordingly. Finally, the system administrator updates the overall book status, and the user can log out. The diagram illustrates the interconnected roles and actions of users, librarians, and system administrators in the library's book borrowing process.

4. Stock Maintenance System

Software Requirement Specification

SRS document for Stock Maintenance System	
Date _____	Page _____
1.1	Introduction
1.1.1	Purpose of this document This document outlines the functional and non-functional requirements for the Stock Maintenance System. It serves as a reference for developers, testers and stakeholders, ensuring alignment on the system's goal.
1.2	Scope of the Document The Stock Maintenance System is designed to automate the tracking and management of stock levels, enabling users to monitor stock levels, manage product orders, deliveries and sales and generate detailed reports on inventory status.
1.3	Overview The Stock Maintenance System will provide real-time updates on stock levels and automate key functions such as low stock alerts, purchase order management, and sales tracking. It will generate reports to help businesses make data-driven decisions regarding inventory management.
2	General Description The Stock Maintenance System is a stand-alone application that can be integrated with existing Point-of-Sale (POS) systems and e-commerce platforms. It is designed to provide inventory management capabilities for businesses of various sizes, with a focus on simplifying stock tracking.
3	Functional Requirements
3.1	Stock Management: + The system should allow users to add, update and delete stock items and should provide real-time stock level updates.

3.1 Order management

- * The system should generate and manage purchase orders for suppliers so that users can track the status of their orders

3.3 Sales Management

- * The system will reduce stock quantities automatically after each sale

3.4 User management: Each user must have a unique login and access must be logged

4 Interface Requirements

- 4.1 User Interface → Web based interface accessible through modern browsers
- 4.2 System Interface → System must support integration with third party POS and ERP Systems via API's

5 Performance Requirements

- 5.1 System must update the stock levels after 3 seconds of sales trade
- * The system must not exceed the error rate of 0.1%

6 Design Constraints

- * Use MySQL database, ReactJS for front end and NodeJS for backend
- * Optimize for use on mobile devices

7 Non Functional Attributes

- ~~Red~~ Security
- * Portability
- * Reliability
- * Usability

Preliminary Schedule

Phase	Description	Duration
Requirements and Planning	Collect requirements and plan the project	2 weeks
System Design	Design the architecture and Interface	3 weeks
Development	Develop core features and functionalities	8 weeks
Testing and Debugging	Perform testing and fix bugs	3 weeks
Deployment and final review	Deploy and finalize	1 week
Total Duration		17 weeks

Budget

Item	Cost Estimate
Development Cost	\$ 30,000
Testing Cost	\$ 10,000
Deployment Costs	\$ 5,000
Maintenance Costs	\$ 5,000 per year
Total Estimated Budget	\$ 45,000

Class Diagram

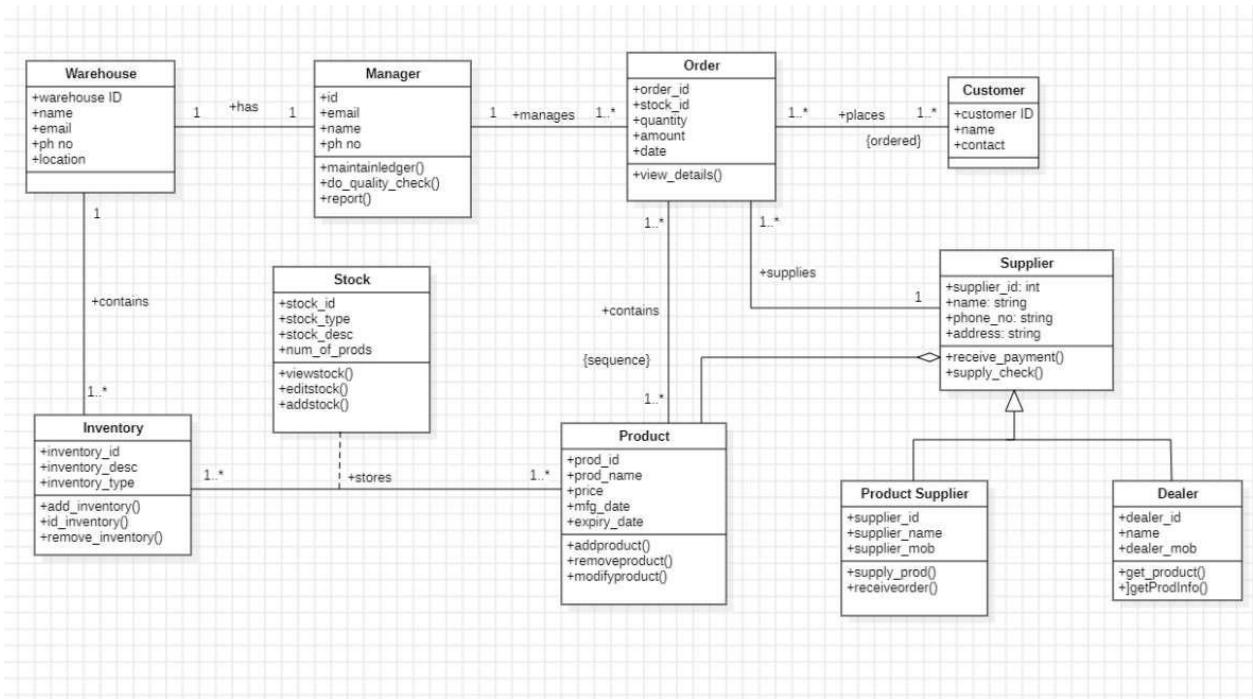


Fig 4.1 Stock Maintenance System - Class Diagram

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

State Diagram

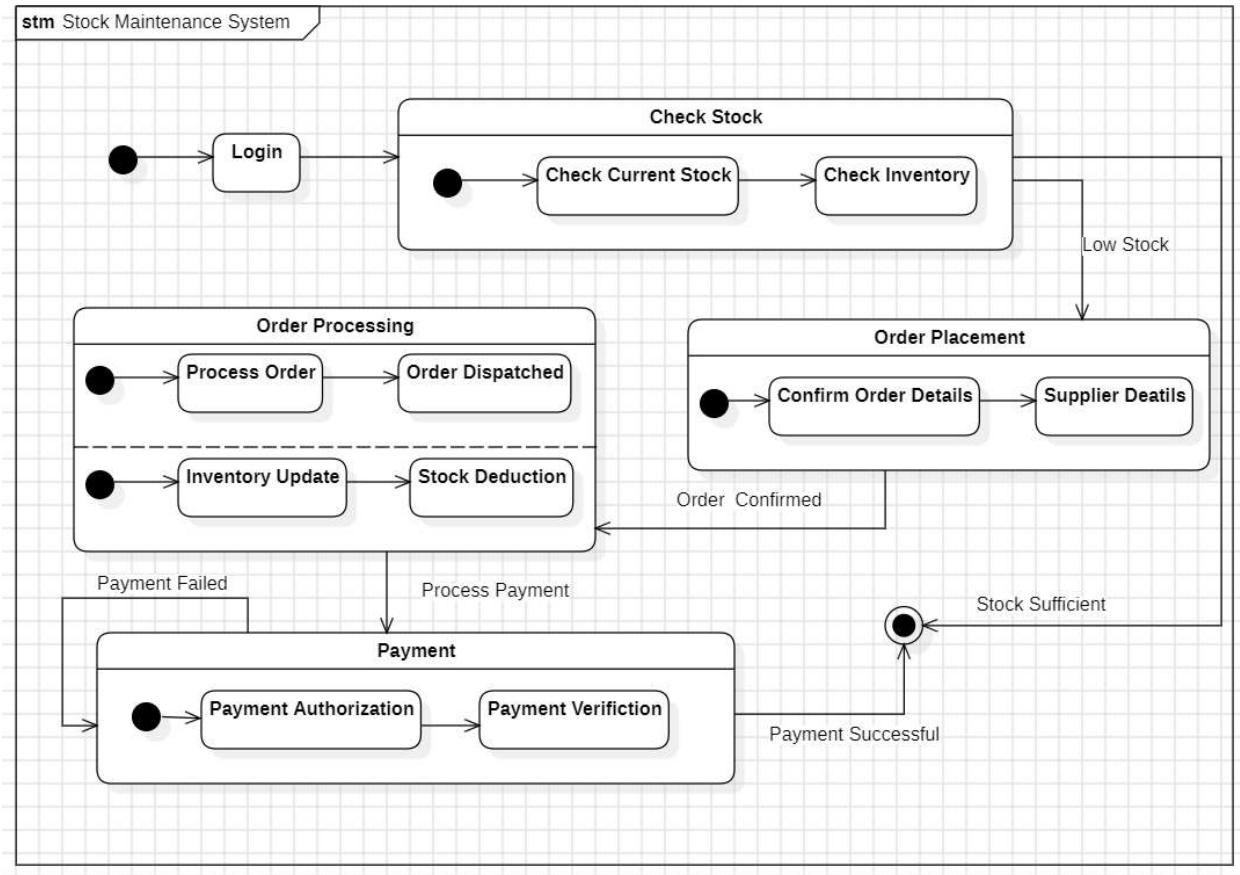


Fig 4.2 Stock Maintenance System - State Diagram

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

Use Case Diagram

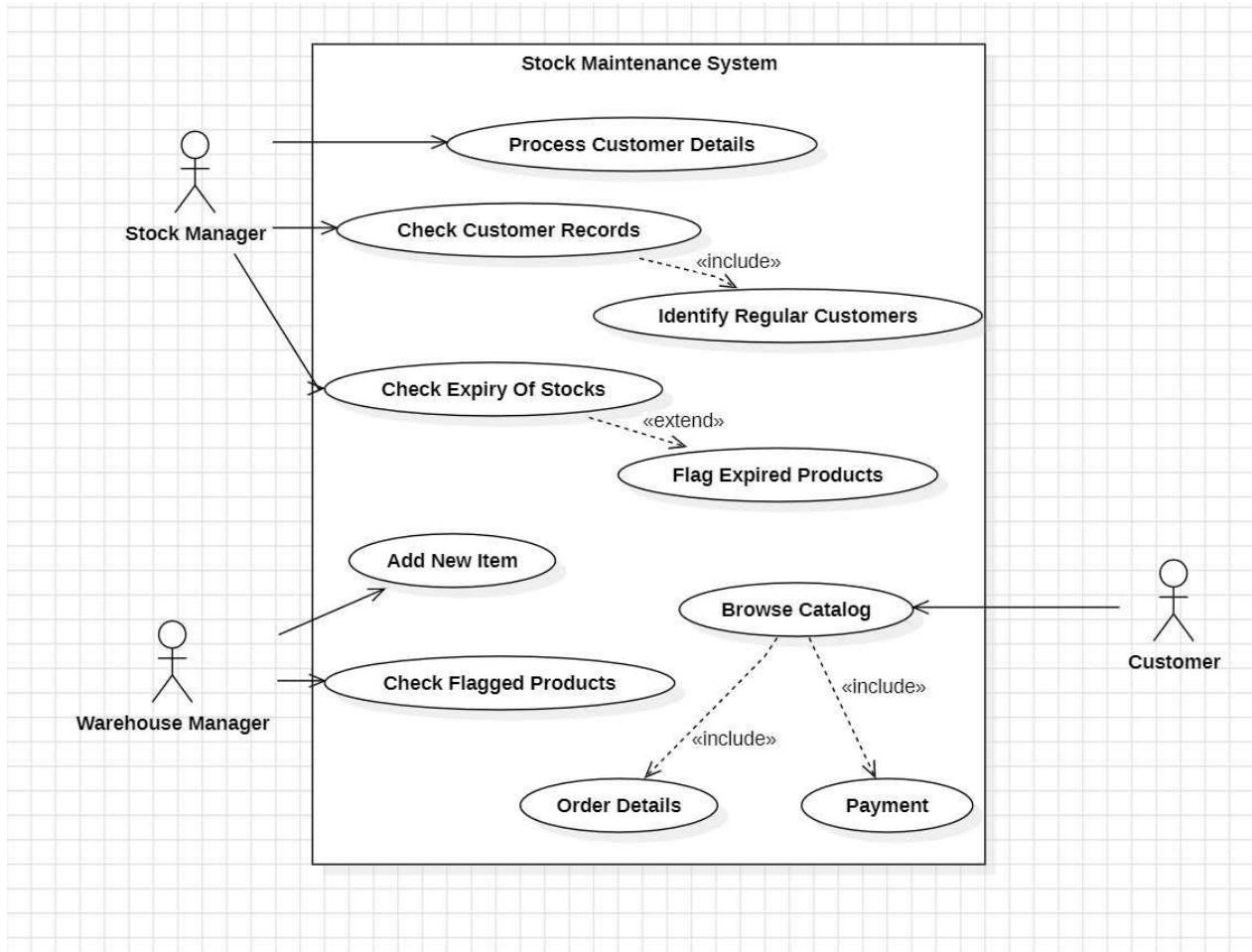


Fig 4.3 Stock Maintenance System - Use Case Diagram

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

Sequence Diagram

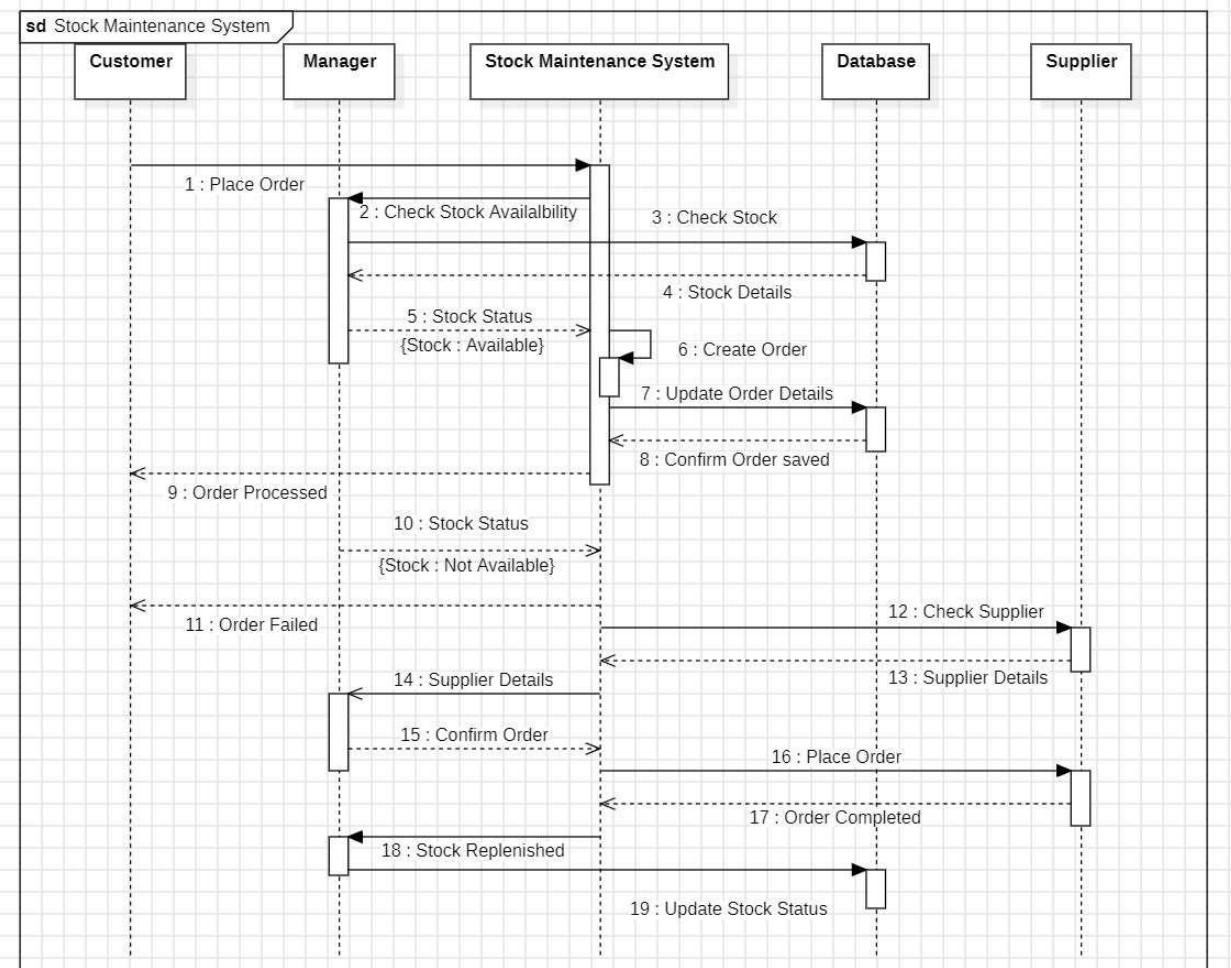


Fig 4.4 Stock Maintenance System - Sequence Diagram

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

Activity Diagram

Fig 4.5 Stock Maintenance System - Activity Diagram

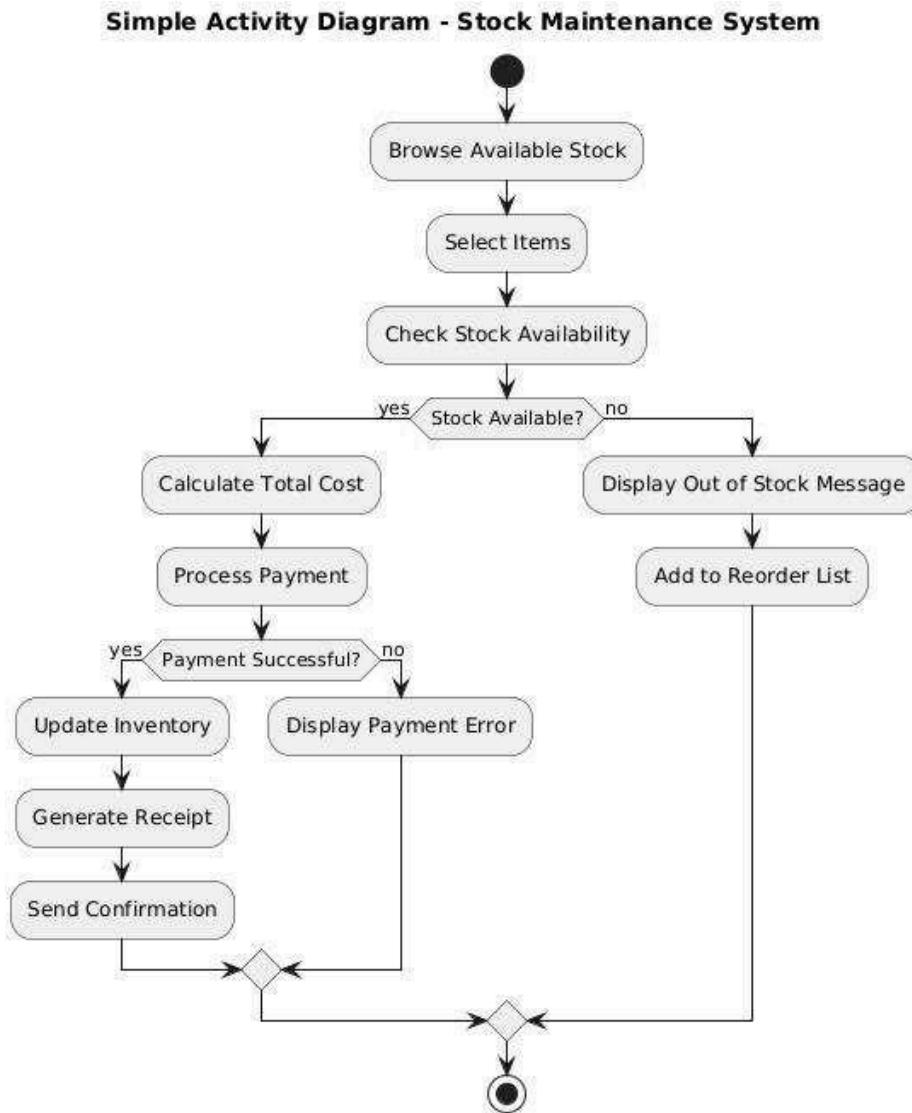


Fig 4.5.1

The stock maintenance process follows a linear flow, including key activities like stock browsing, availability checks, payment processing, inventory updates, and receipt generation. Decision points include stock availability and payment success/failure, ensuring simplicity and clarity for stakeholders while focusing on core functionality.

5. Passport Automation System

Software Requirement Specification

classmate
Date _____
Page _____

SRS document for Passport Authentication System

1.1 Introduction

1.1.1 Purpose of this Document

The purpose of this document is to define the functional and non-functional requirements for a passport authentication system. It serves as a detailed guide for developers, testers and stakeholders to ensure a clear understanding of the system's goal, scope and features.

1.1.2 Scope of the System

The Passport Authentication System will provide a secure method for verifying the authenticity of passports by cross-referencing the passport data with a global or national database. The system will reduce the risk of passport fraud, improve border security, and streamline the process of identifying and verifying individuals.

1.1.3 Overview

The Passport Authentication System is a software solution designed to authenticate passports in real time by scanning and validating passports against a secure, centralized data base. The system will allow authorized users to scan a passport, verify its credibility, and validate it instantly.

2 General Description

2.1 ~~Possible~~ The Passport Authentication System will function as a standalone application, but will also be capable of integration with existing border security and immigration systems. It will be designed to ensure fast and accurate authentication of passports at border checkpoints, airports and other secure locations.

3 Functional Requirements

- * Passport Scanning and Data Capture
The system must capture from the MRZ (Machine Readable Zone) using a compatible scanner.
- * Passport Data Validation
- * Fraud Detection
- * Reporting and logs
- * etc

4 Interface Requirements

- * User Interface
- * System Interface

5 Performance Requirements

- * Must authenticate Passport in 3-5 seconds
- * System must maintain accuracy rate of over 99.9%

6 Design Constraints

- * The system should be built using a robust and secure database.
- * It must comply with the International Security standards.

7 Non Functional Attributes

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

Preliminary Schedule and Budget

Phase	Description	Duration
1 Requirements gathering and planning	Collect system requirements and plan the project	3 weeks
2 System Design	Design the architecture and user interface	4 weeks
3 Development	Develop core functionalities and system integration	10 weeks
4 Testing and Debugging	Perform system testing and bug fixing	4 weeks
5 Deployment and Final Review	Deploy the system and review final deliverables	2 weeks
Total Duration		23 weeks

Preliminary Budget

Item	Cost Estimate
Development Costs	\$ 50,000
Testing Costs	\$ 15,000
Deployment Costs	\$ 10,000
Maintenance Costs (Yearly)	\$ 8,000
Total Estimated Budget	\$ 75,000

Class Diagram

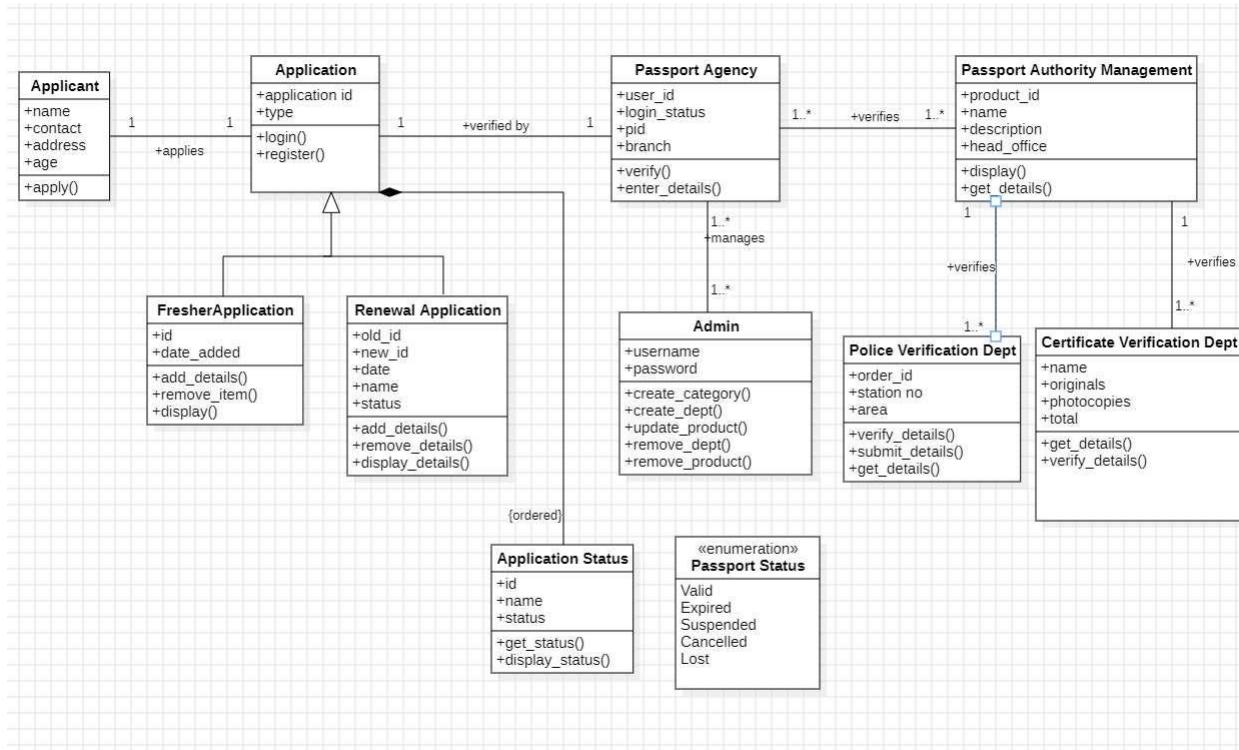


Fig 5.1 Passport Automation System - Class Diagram

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 **Renewal Application**. 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

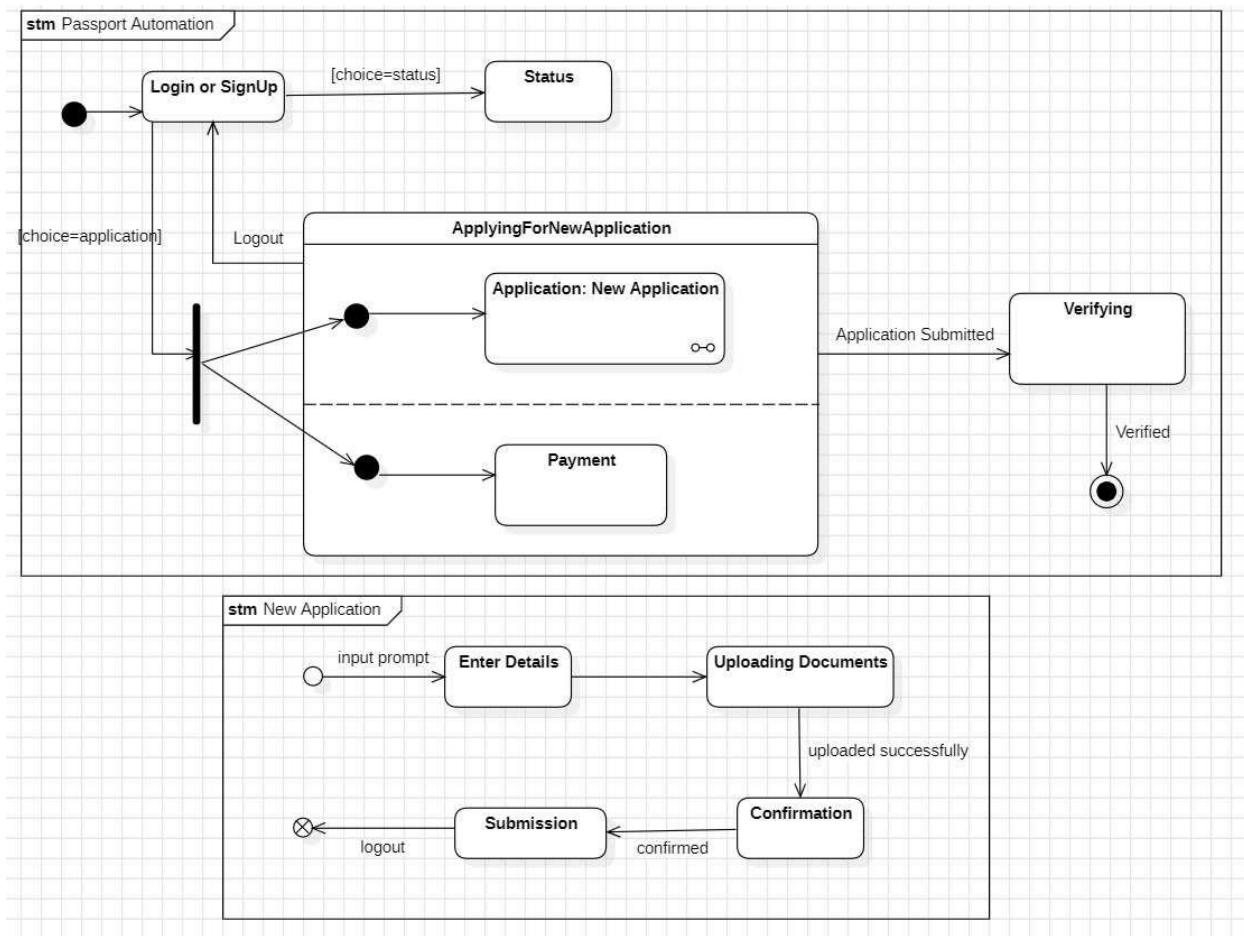


Fig 5.2 Passport Automation System - State Diagram

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

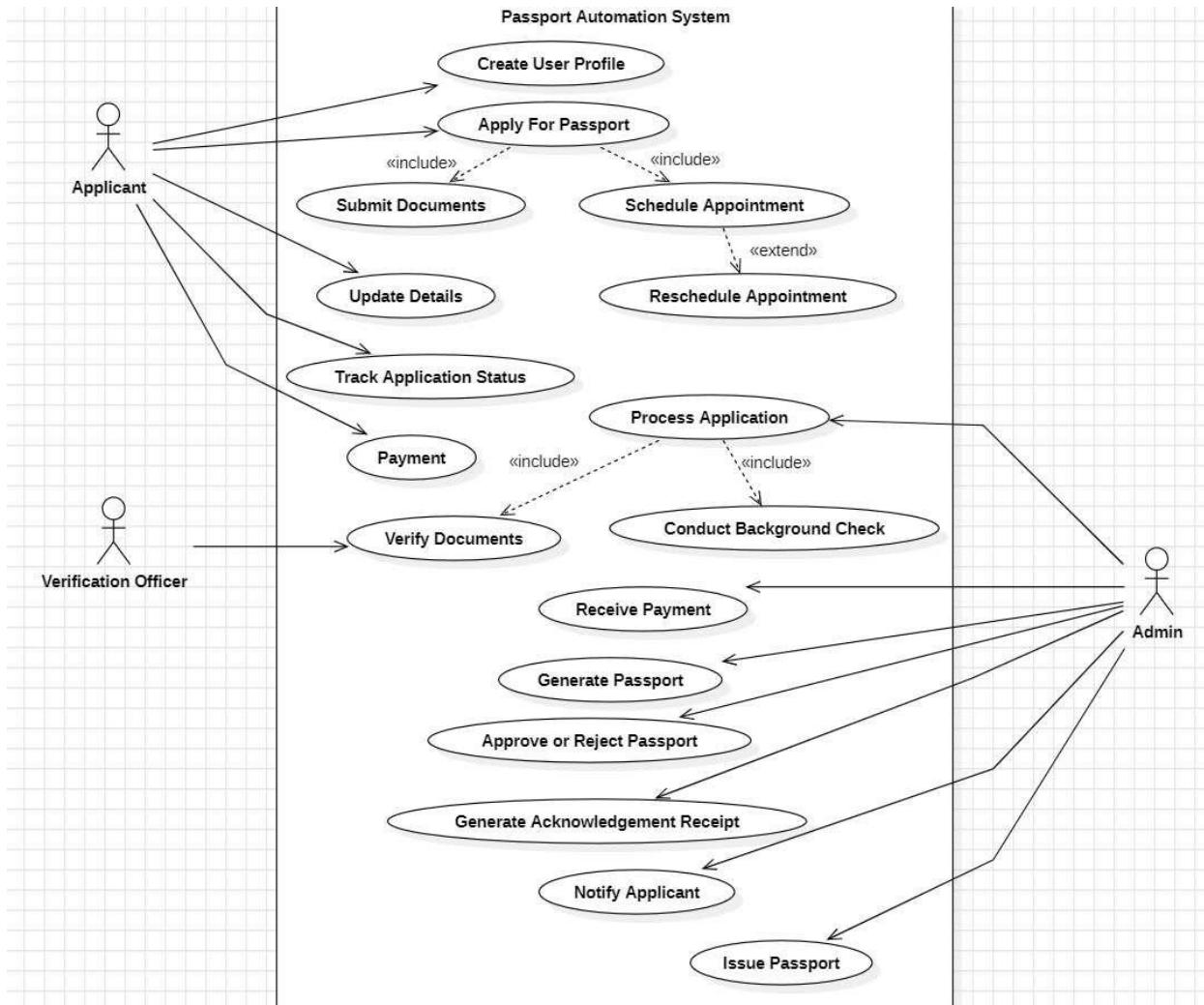


Fig 5.3 Passport Automation System - Use Case Diagram

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

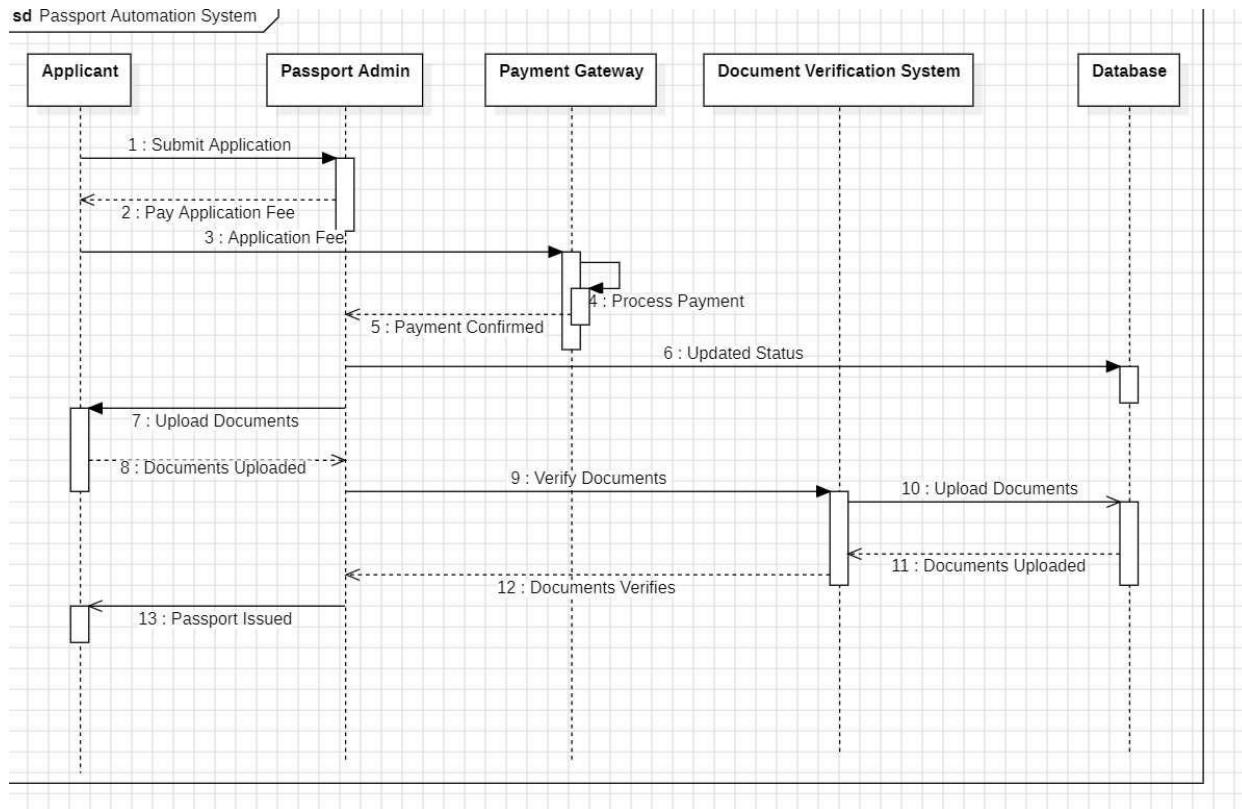


Fig 5.4 Passport Automation System - Sequence Diagram

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

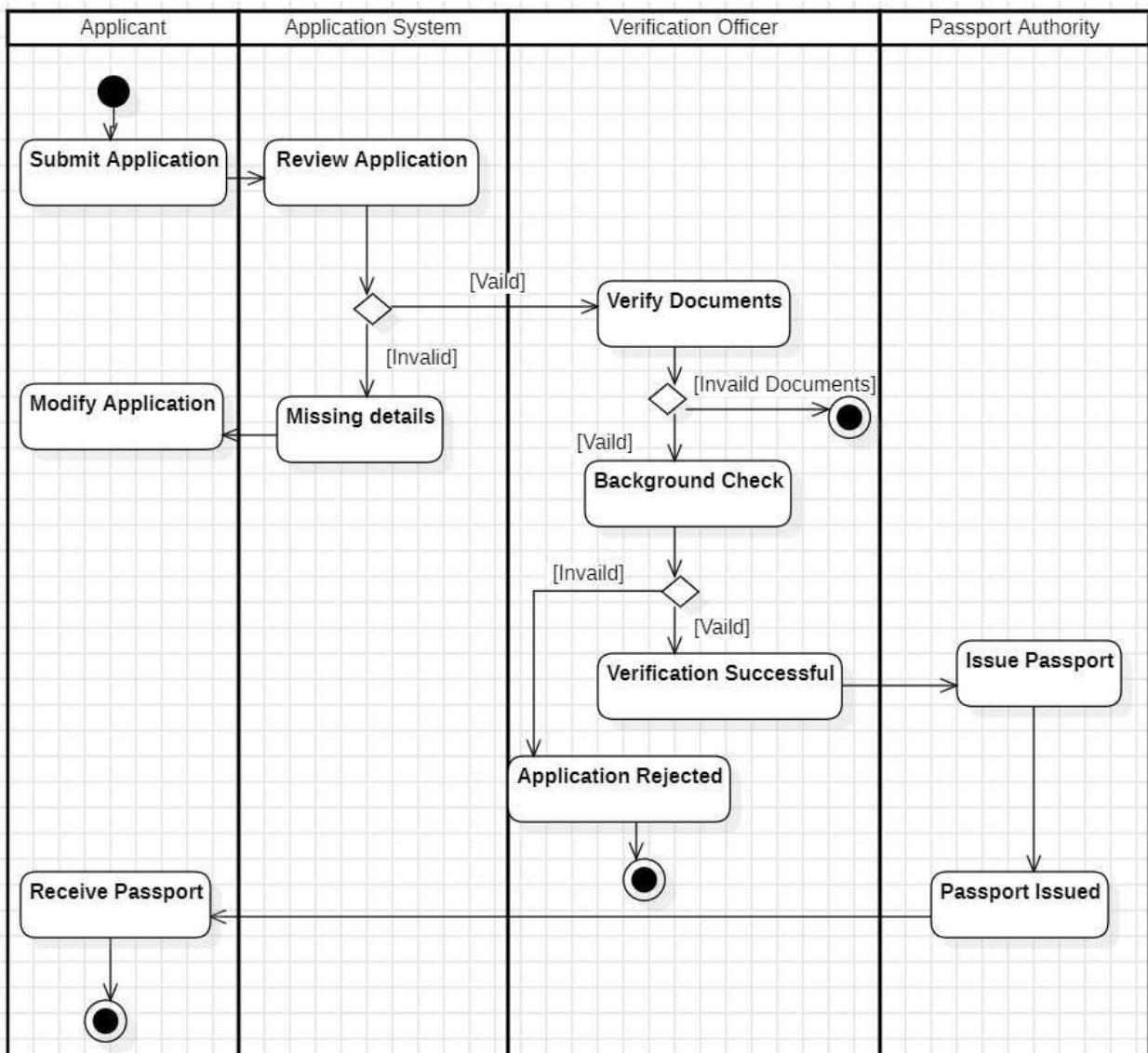


Fig 5.5 Passport Automation System - Activity Diagram

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.