

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on
OBJECT ORIENTED MODELING

Submitted by

Sanjana Shetty
(1BM22CS238)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING

in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
September-2024 to January-2025

**B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED MODELING**" was carried out by **Sanjana Shetty (1BM22CS238)**, who is a bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-2025. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Modeling-(23CS5PCOOM)** work prescribed for the said degree.

M Lakshmi Neelima
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Kavitha Sooda
Professor and Head
Department of CSE
BMSCE, Bengaluru

Table of Content

Sl. No.	Experiment Title	Page No.
1	Hotel Management System	1-9
2	Credit Card Processing	10-17
3	Library Management System	18-24
4	Stock Maintenance System	25-31
5	Passport Automation System	32-38

https://github.com/sanjanashettybmsce/oomd_1BM22CS238

Hotel Management System

Software Requirements Specification (SRS)

SRS - Hotel Management

1. Introduction

1.1 Purpose of the document

To understand working and functionalities of HMS (Hotel Management System) which will benefit developers and stakeholders. The document will include requirements in all aspects and provide complete details of the hotel management system.

1.2 Scope of this document

The document covers detailed estimations of cost and time in order to understand all schedules and budgets associated to hotel management system.

1.3 Overview

The SRS document will cover all kinds of requirements (functional, non-functional, performance, interface, requirements) along with design constraint and scheduling & budgeting.

2. General description

The Hotel management system (HMS) shall include various facilities such as bookings/reservation system, room services, facilities, offers/discounted prices as a few examples. Facilities and services can be provided in-person or online via web applications and mobile apps.

3. Functional Requirements

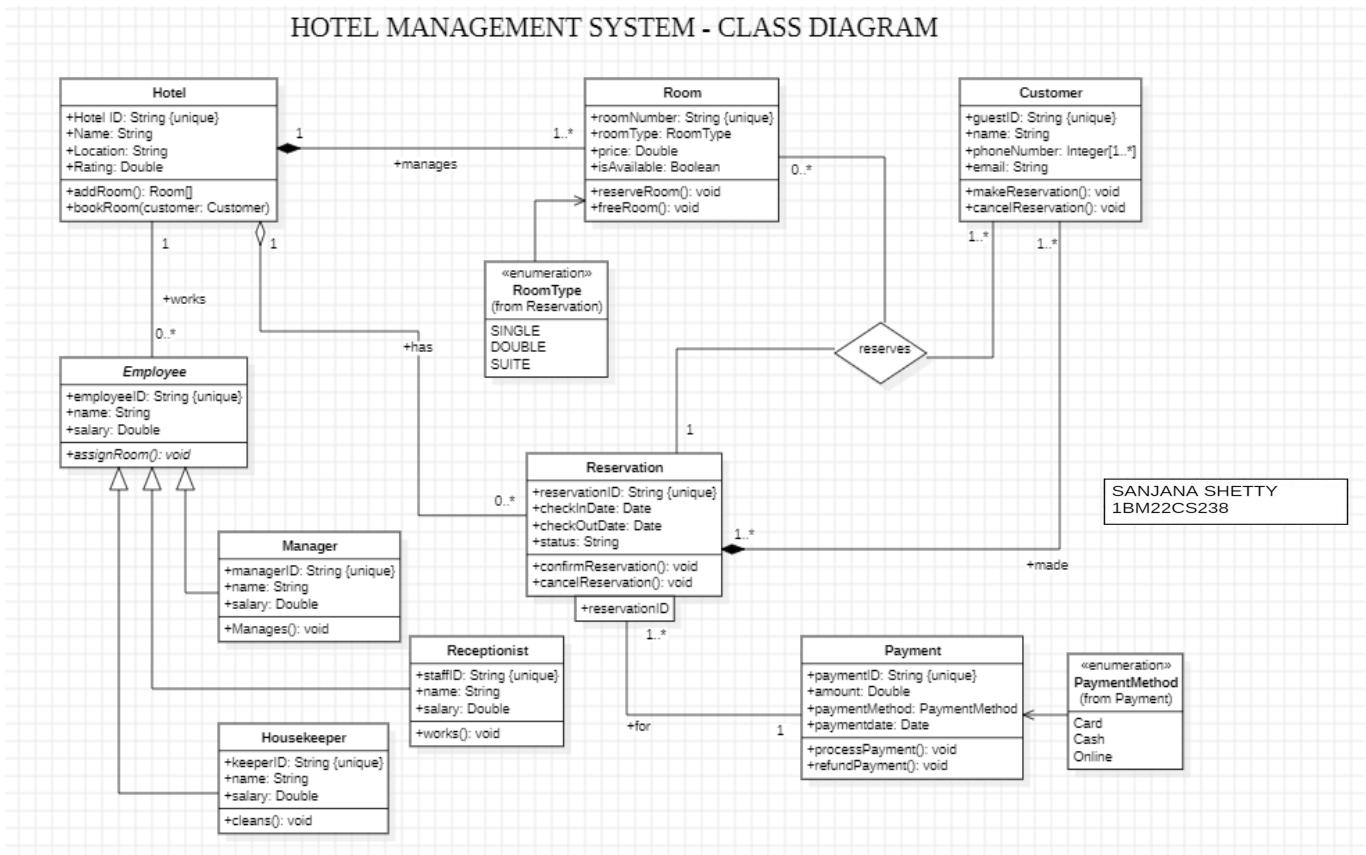
- functional requirements include:
 - a) providing users with option to book reservations in the hotel (Booking management)
 - b) providing guests with room services and other facilities (services) - guest management
 - c) guest can avail offers/discounts on various choices of the hotel or during the hotel booking process. (via web application)
 - d) Provision of room services by the hotel manager/worker
 - e) Provision of club memberships by hotels for sports facilities & other miscellaneous advantages.
 - f) Providing food (meals) such as breakfast, lunch, dinner with a menu driven application in hotel app for the same to make placing orders easier.

CLASSMATE	
Date	Page
4.	<u>Interface Requirements</u>
	An interface that can be made available to users will be a mobile application through which users can make bookings, reserve rooms and apply/renew club memberships.
	for The mobile application can have a user friendly interface and smooth navigation to ensure successful transactions at all times.
5.	<u>Performance Requirements</u>
	One of the basic requirements will be a fast working mobile application with the ability to work well with a large load of users too (5000 users at max). The mobile application must have a majority of successful transactions else it should revert back to previous state completely.
6.	<u>Design Constraints</u>
	a) If a certain transaction (such as, payment) take too long to process (more than 4 minutes), then the payment/transaction must be aborted and rollback to its previous state.

6)	The UI of the website should align with the branding of the hotel and must cater to the colour scheme and logo too.
7)	
2.	<u>Non-functional attributes</u>
a)	Speed & durability of the system must be good and efficient.
b)	Portability of goods & services to and from the hotel.
c)	Security of user data must not be breached. Hence, a secure database technology must be used.
d)	Legal corporations associated with the company can help with legal documentations & other authorizations.
e)	Hiring of workers specifically hotel management graduates. Hiring a team of workers to work under manager & receptionists too.
f)	Creation of domains (ensuring that a manager heads each of them).
g)	Authenticating users of hotel management system.

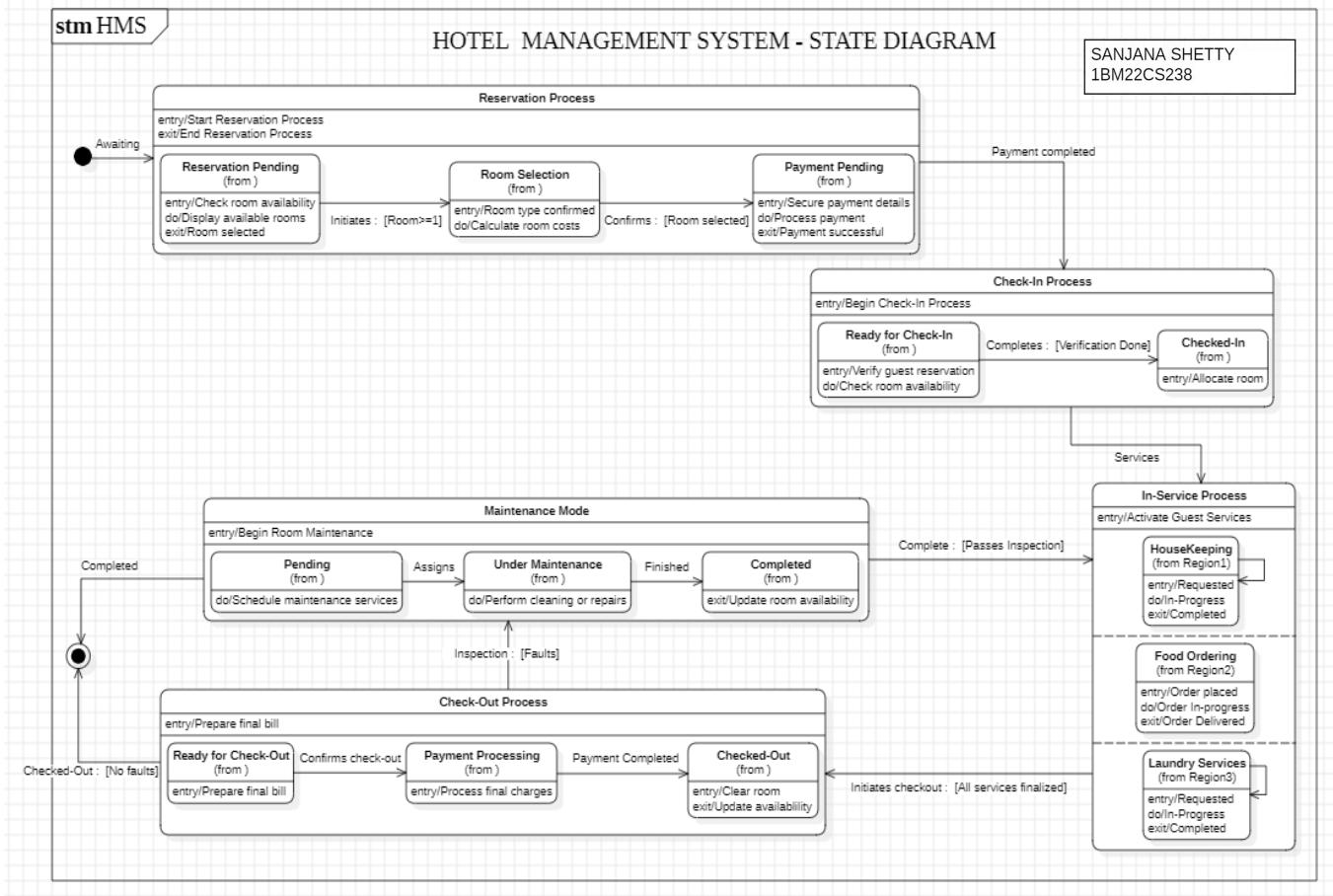
- classmate
Date _____
Page _____
- 8- Preliminary Schedule & Budget
- a) All reservations & bookings must be specified in a timeline to keep track of simultaneous bookings.
 - b) Budgeting must be done with financial advisors!
- N/A

CLASS DIAGRAM



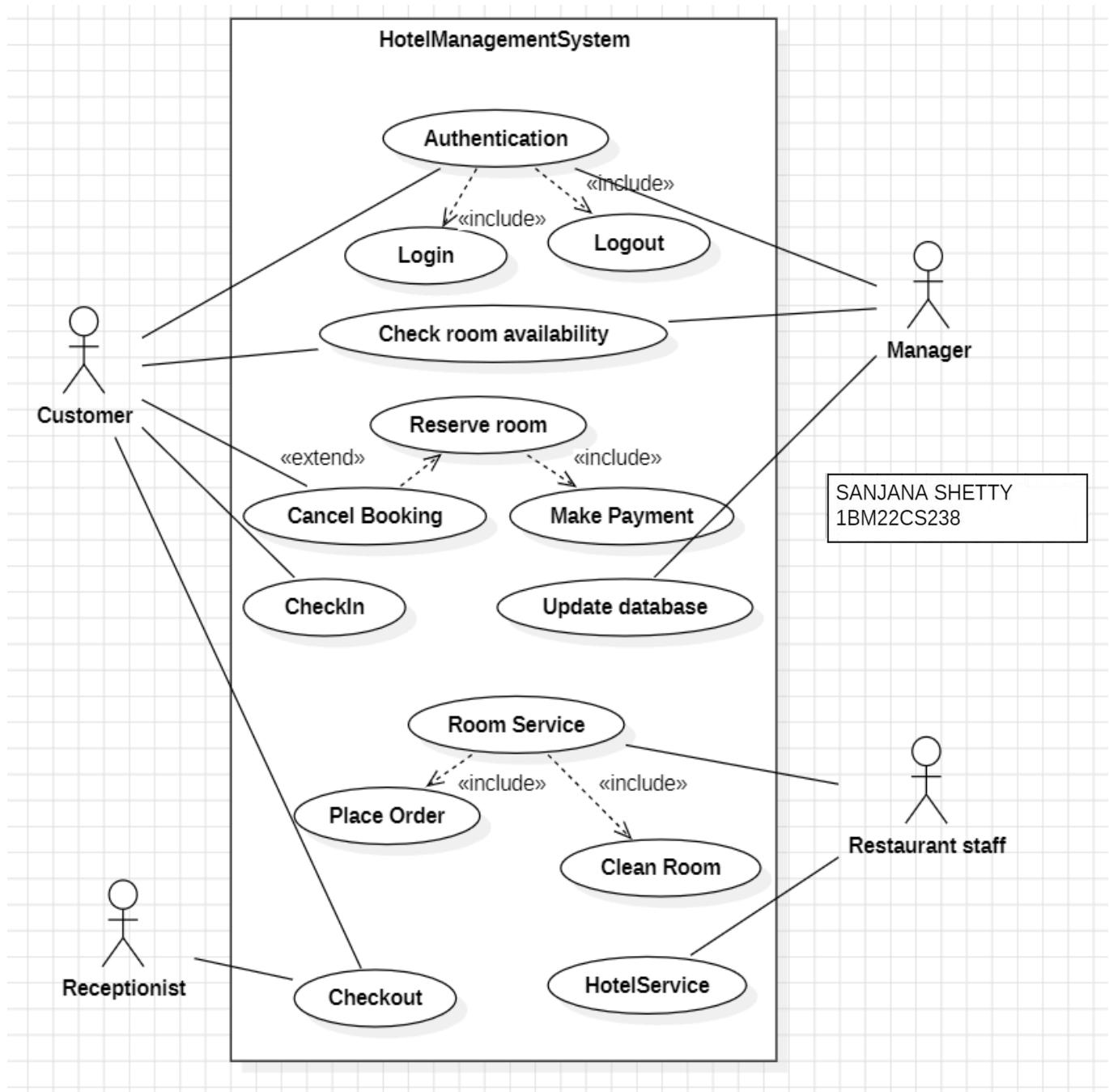
The class diagram illustrates a Hotel Management System comprising various entities and their relationships. The central class, Hotel, manages multiple Rooms, each identified by attributes like roomNumber, roomType, price, and availability. Customers can reserve rooms through the Reservation class, which records details like checkInDate, checkOutDate, and status, and is associated with Payment, handling methods such as processPayment() and refundPayment(). The system includes Employees, categorized as Managers, Receptionists, and Housekeepers, each with specific responsibilities such as managing operations, handling reservations, or maintaining cleanliness. The Customer class allows users to make or cancel reservations. Enumerations for RoomType (e.g., SINGLE, DOUBLE) and PaymentMethod (e.g., Card, Cash) add structured classifications. Overall, the diagram captures the relationships and functionalities necessary for a comprehensive hotel management system.

STATE DIAGRAM



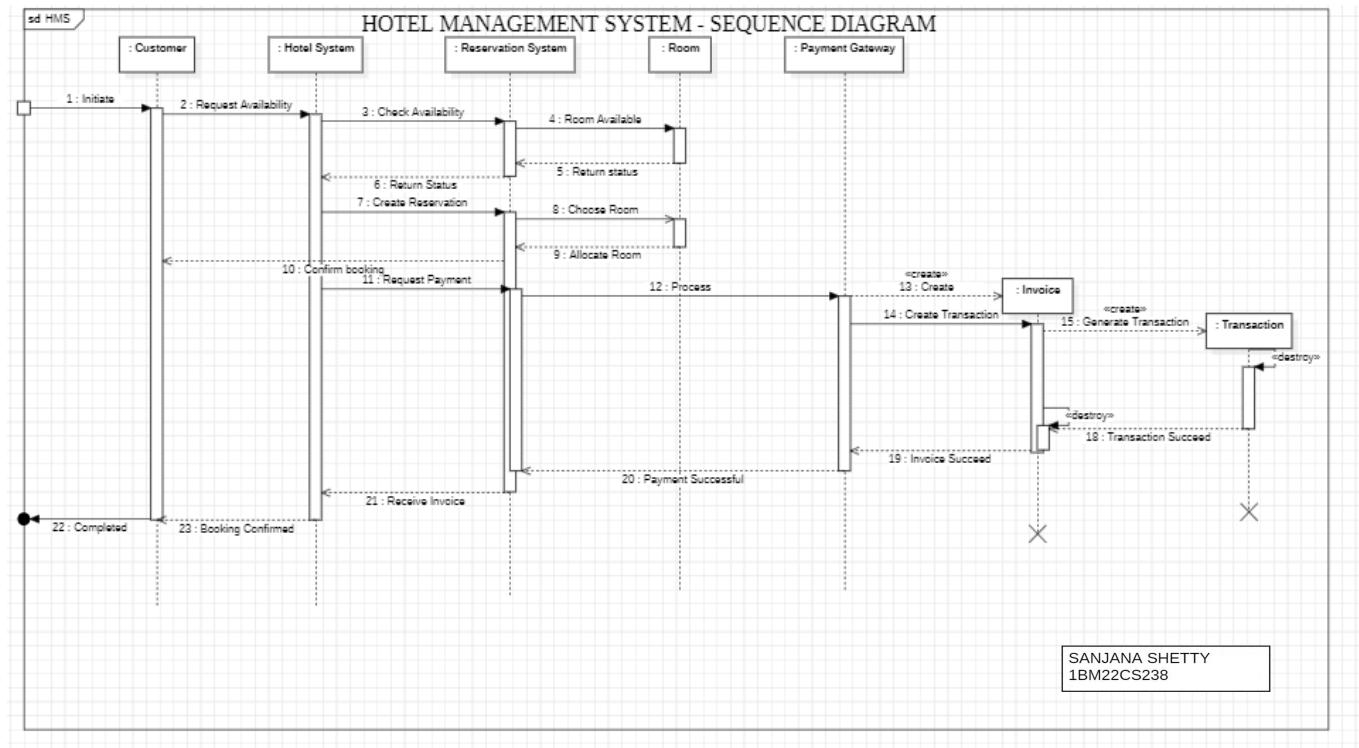
The state diagram depicts the Hotel Management System processes, starting with the Reservation Process (from Reservation Pending to payment completion), followed by the Check-in Process (Ready for Check-In to Checked-In). During the stay, the In-Service Process manages tasks like Housekeeping, Food Ordering, and Laundry Services. Simultaneously, Maintenance Mode ensures room upkeep (Pending to Completed). The system concludes with the Check-Out Process, including billing, payment, and clearing rooms. The diagram effectively outlines the workflow and transitions between states.

USE CASE DIAGRAM



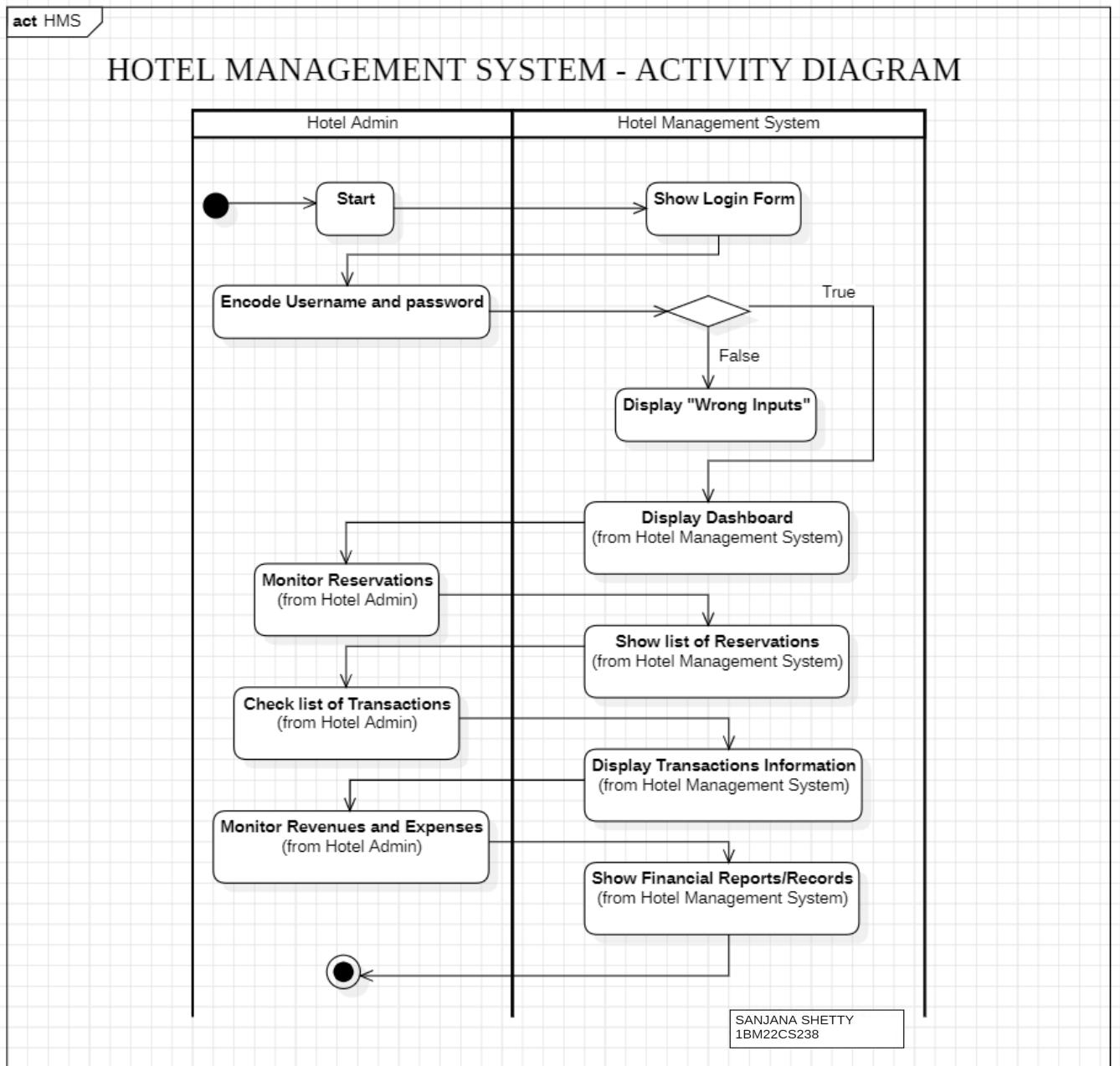
The use case diagram depicts the Hotel Management System, highlighting interactions between Customer, Manager, Receptionist, and Restaurant Staff. Key actions include Authentication, Room Reservation, Payment, Check-in, Room Service, and Checkout. It shows relationships like include and extend to represent interconnected functionalities within the system.

SEQUENCE DIAGRAM



The diagram illustrates the sequence of interactions involved in a hotel booking process. It starts with a customer initiating the request, followed by the hotel system checking availability and returning the status to the customer. Upon confirmation, the reservation system creates a reservation and allocates a room. The customer then proceeds to request payment, which is processed by the payment gateway. Once the payment is successful, an invoice is created and a transaction is generated. Finally, the customer receives the invoice and the booking is confirmed.

ACTIVITY DIAGRAM



The diagram illustrates the activity flow of a hotel management system. It begins with the hotel admin starting the system and being presented with a login form. After successfully entering their credentials, the admin is granted access to the dashboard. From there, the admin can perform various tasks, such as monitoring reservations, checking transactions, and viewing financial reports. The system provides the admin with the necessary information and tools to manage these aspects of the hotel.

CREDIT CARD PROCESSING

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

3/10/24 SRS for Credit Card Processing system

1. Introduction

1.1 Purpose of the Document

The purpose of this SRS document is to outline the functional and non-functional requirements of the system. The document will cover all details of the credit card processing system.

1.2 Scope of this document

The document will cover details such as estimation costs, development costs, user experience, security and efficiency. It will also include timelines for project planning and guidelines.

1.3 Overview

The document will cover all aspects of the credit card system. The system will ensure accurate transaction processing, fraud detection and compliance with industry regulations to enhance customer satisfaction.

2 General description

The primary features and functions include transaction processing for secure and accurate money transfers, fraud detection to ensure there is no fraudulent use and reporting to merchants and users of the credit card processing system.

3 Functional Requirements

1. User Registration - users are registered with the credit card.
2. Transaction Initiation - customers initiate a transaction using credit card information.
3. Transaction Validation - System must validate credit card details.
4. Payment Authorization - transactions should be authorized from payment gateway.
5. Fraud Detection Algorithm - usage of fraud detection technology for security.
6. Transaction Reporting - reporting transaction details to all parties.

2. General description

The primary features and functions include transaction processing for secure and accurate money transfers, fraud detection to ensure there is no fraudulent use and reporting to merchants and users of the credit card processing system.

3. Functional Requirements

1. User Registration - users are registered with the credit card.
2. Transaction Initiation - customers initiate a transaction using credit card information
3. Transaction Validation - System must validate credit card details
4. Payment Authorization - transactions should be authorized from payment gateway
5. Fraud Detection Algorithm - usage of fraud detection technology for security
6. Transaction Reporting - reporting transaction details to all parties

4. Interface

a) Payment Gateway Interface : communication with external payment processors via APIs

b) User Interface : a front-end in the form of a website to interact with the user for transactions

c) Database Interface :
a database to store & retrieve transaction data

5. Performance Requirements

a) Performance - response time with respect to transaction processing within 2 seconds under normal conditions

b) Scalability - the system should support upto 1000 simultaneous transactions

c) Uptime - service should be continuously available at a rate of 99.9%

6. Design Constraints

- a) Regulatory Compliance:
The system must adhere / be compliant to Data Security regulations.
- b) Technology Stack:
The system must utilize programming languages like Java and frameworks such as Node.js.
- c) Hardware Specifications:
Must operate on specific configurations (minimum RAM and CPU of OS).

7. Non-Functional Attributes

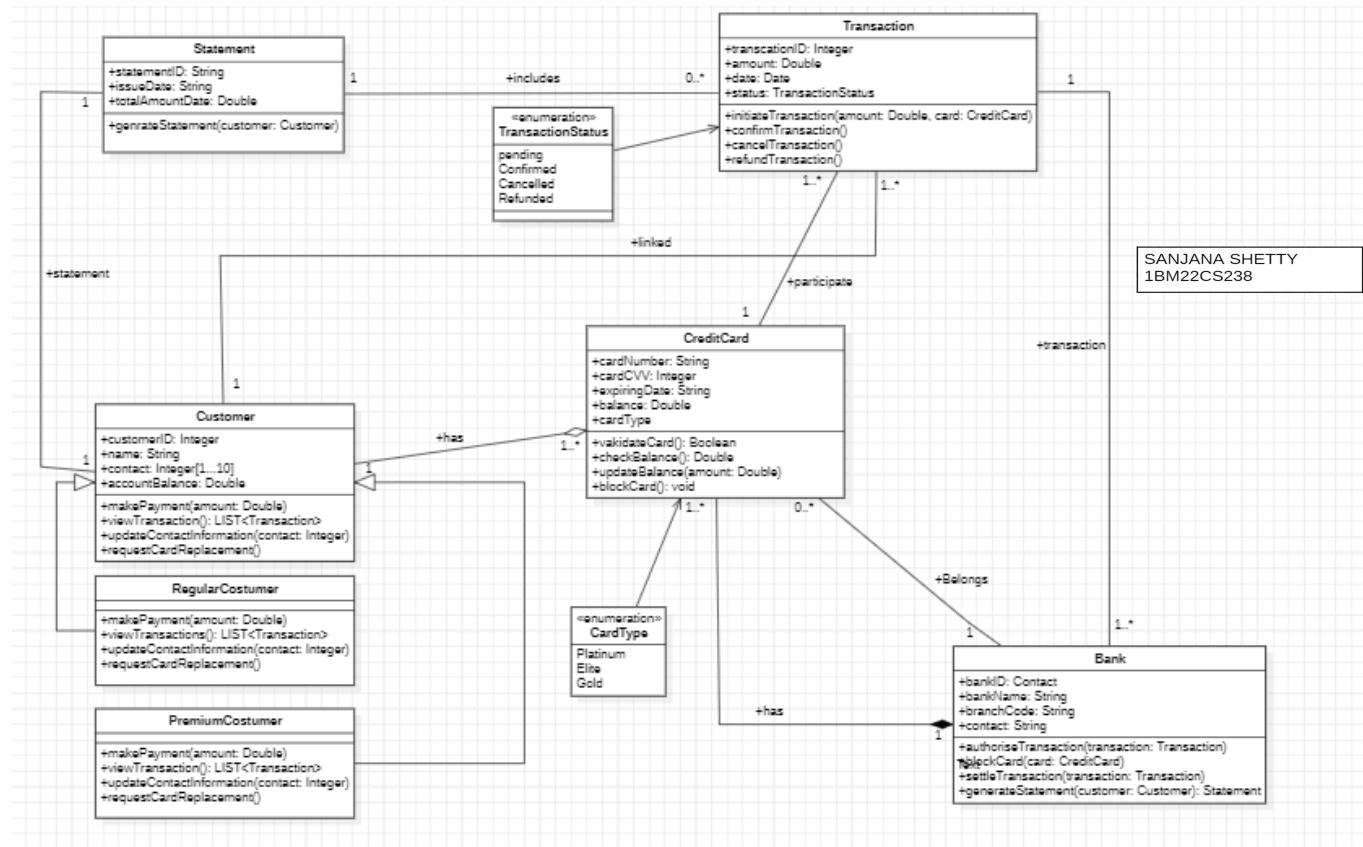
- a) Security: data encryption and secure transaction protocols.
- b) Reliability: should have minimal downtime.
- c) Portability: compatible with multiple operating systems and multiple configurations of RAM.
- d) Scalability: ability to handle loads as number of transactions increase.

8. Preliminary Schedule and Budget

Estimated project duration: 6 months

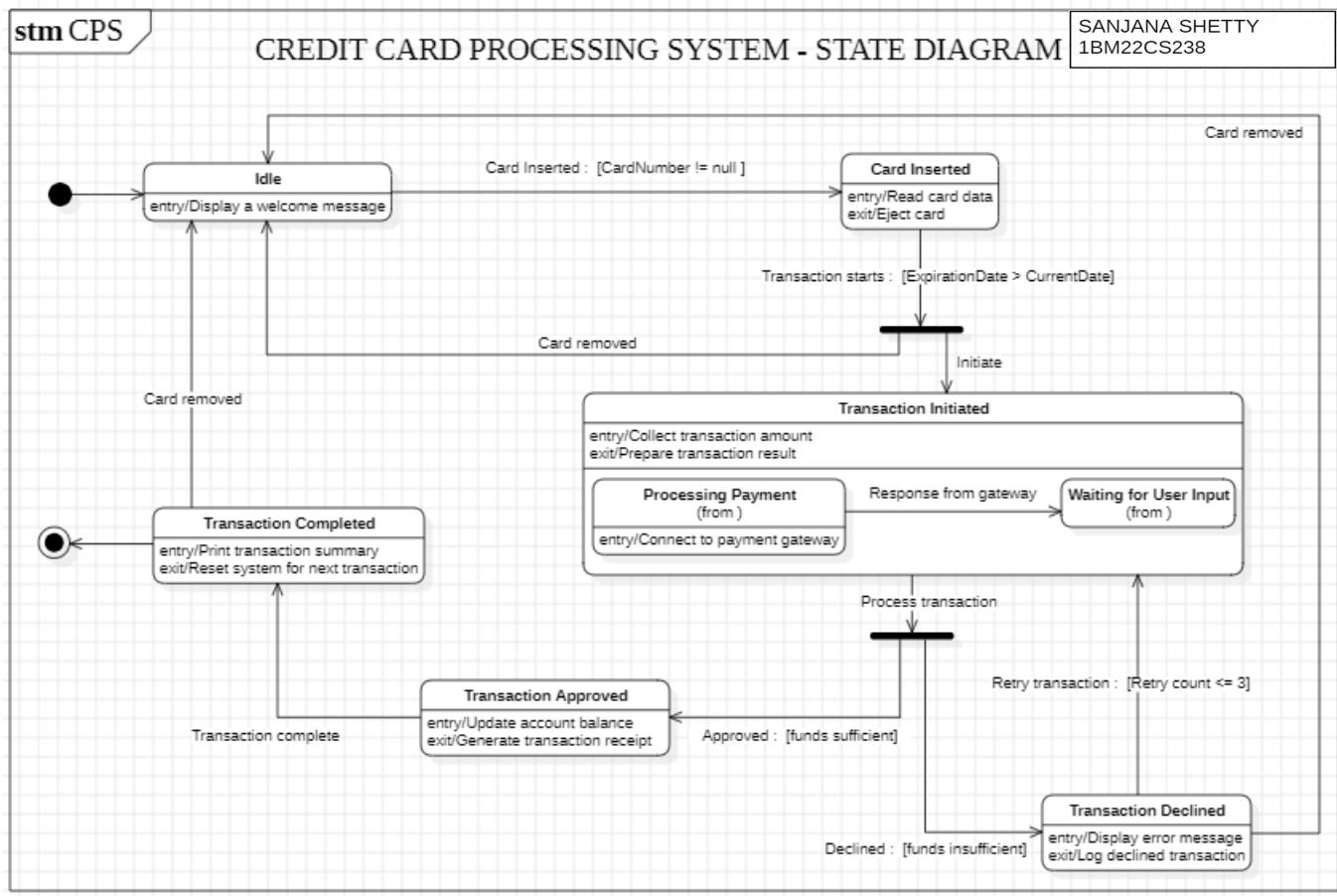
Budget: overall cost estimation is Rs. 200,000

CLASS DIAGRAM

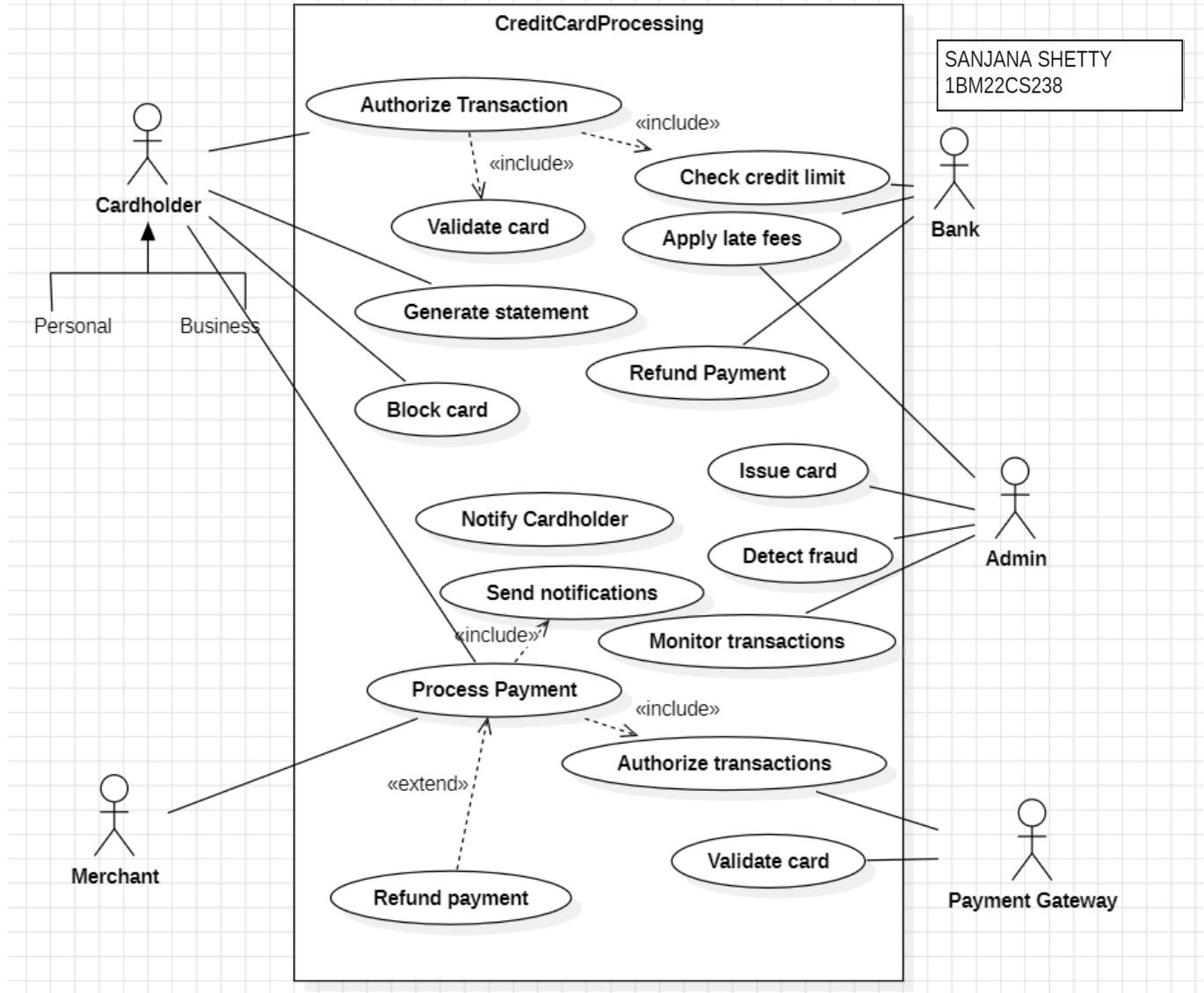


This UML class diagram illustrates the core components of a financial system. It depicts entities such as Customer, CreditCard, Transaction, Statement, and Bank, along with their attributes and relationships. Key features include transaction management, statement generation, and customer categorization based on their spending habits. The diagram also includes enumerations for transaction status, card type, and customer rank.

STATE DIAGRAM

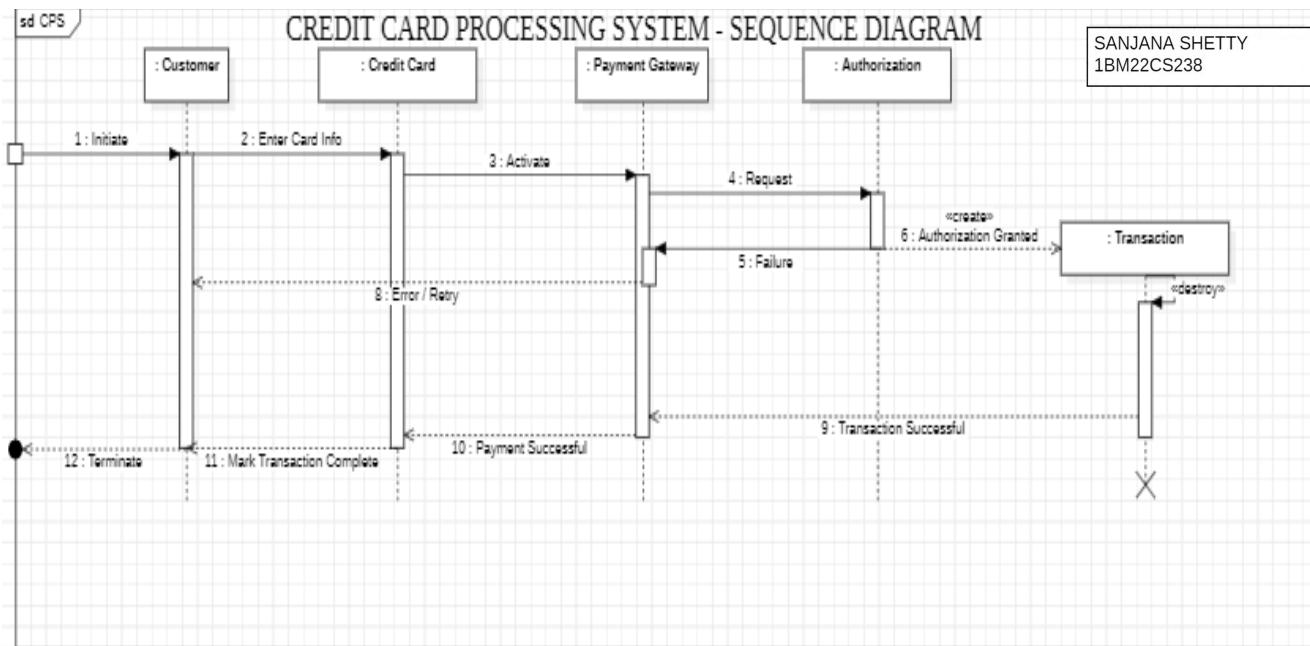


USE CASE DIAGRAM



This UML use case diagram outlines the functionalities of a CreditCardProcessing system. It shows various actors like Cardholder (Personal and Business), Bank, Admin, Merchant, and Payment Gateway interacting with the system through different use cases. Key features include authorizing transactions, validating cards, processing payments, issuing cards, detecting fraud, generating statements, and managing refunds. The diagram also utilizes relationships like <<include>> and <<extend>> to represent dependencies between use cases.

SEQUENCE DIAGRAM

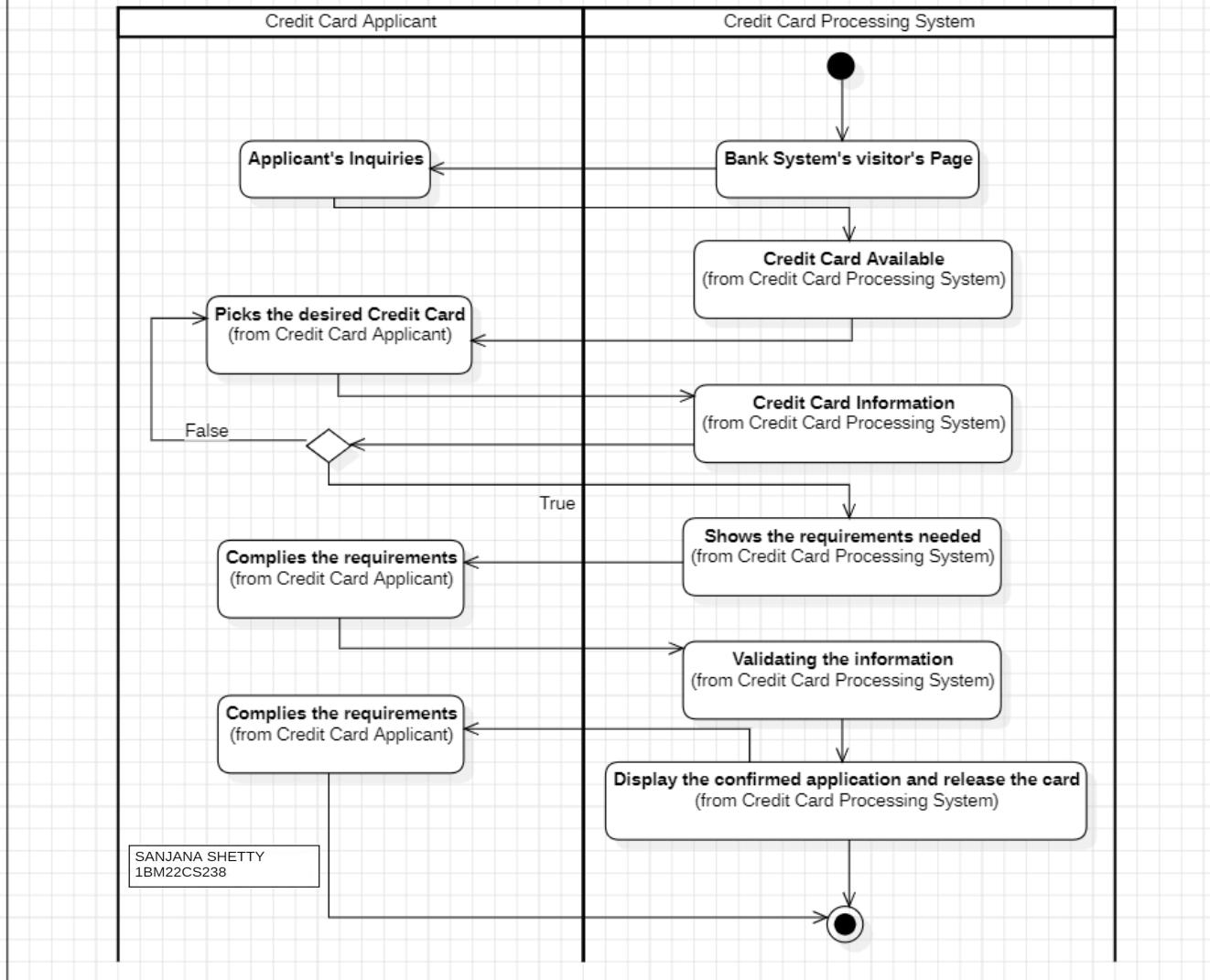


This UML use case diagram models the functionalities of a credit card processing system. It shows how various actors, including cardholders, banks, merchants, and administrators, interact with the system. Key use cases include authorizing transactions, validating cards, processing payments, issuing cards, detecting fraud, generating statements, and managing refunds. The diagram also incorporates relationships like <<include>> and <<extend>> to represent dependencies between use cases, providing a more comprehensive view of the system's behavior.

ACTIVITY DIAGRAM

act CCPS

CREDIT CARD PROCESSING SYSTEM - ACTIVITY DIAGRAM



This activity diagram models the process of a credit card application. It starts with the applicant making inquiries, followed by the system displaying available credit card options. The applicant then selects a desired card. The system checks if the applicant meets the requirements for the selected card. If they do, the system validates the information provided by the applicant. Finally, if the validation is successful, the system confirms the application and releases the credit card. If the applicant fails to meet the requirements at any stage, the system displays the missing requirements.

LIBRARY MANAGEMENT SYSTEM

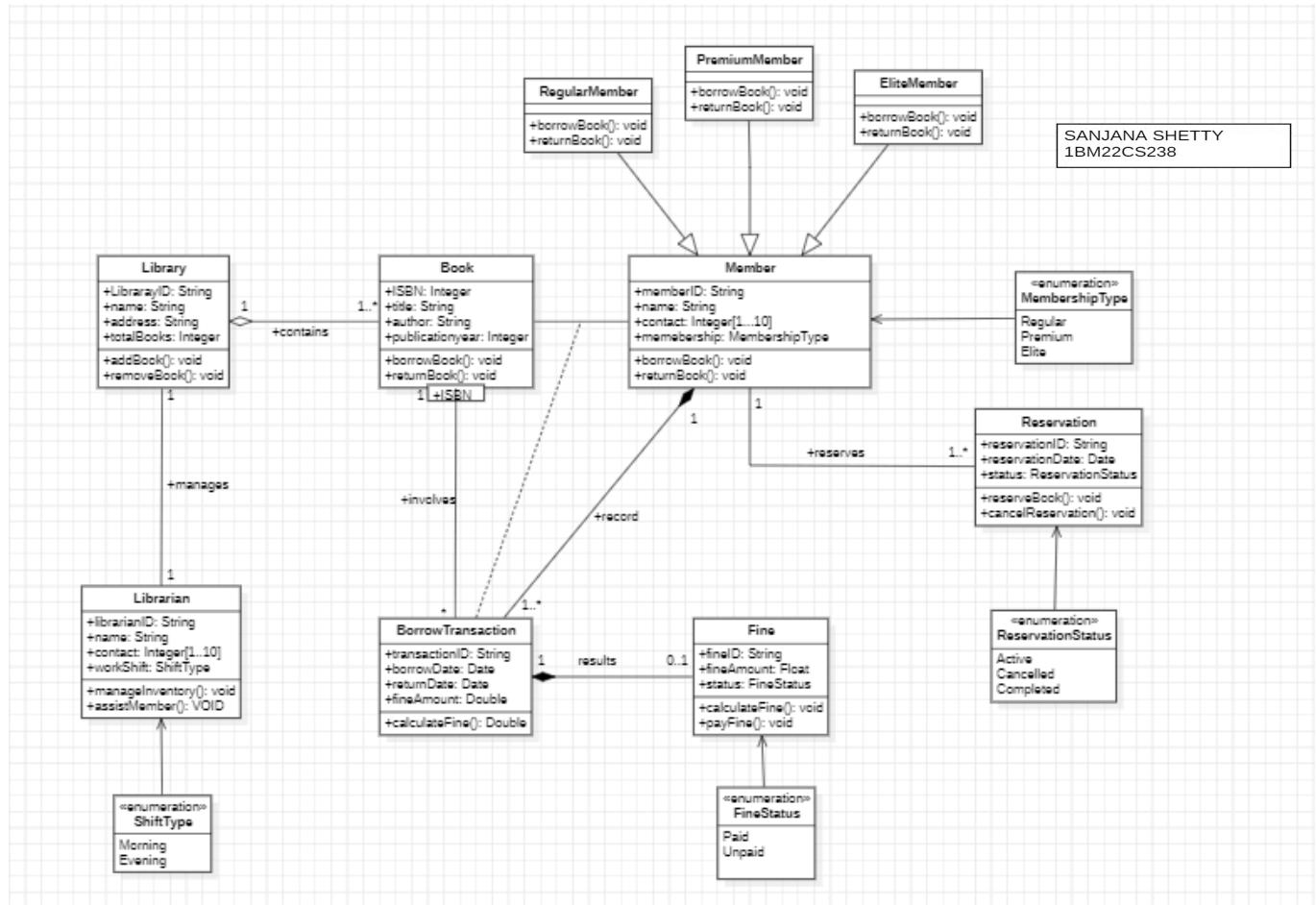
SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

SRS for Library Management System (LMS)	
1. Introduction	1.1 Purpose of this Document The purpose of this document is to cover information with respect to functional and non-functional requirements. The document will cover all the specifications regarding a Library Management System (LMS).
1.2 Scope of this document	The document will outline objectives and functionalities of the LMS detailing information on library staff, users and administrators. Estimation on development costs and timelines is also specified.
1.3 Overview	The LMS is designed to automate processes of borrowing, returning and issuing books. It aims to enhance user experience, resource tracking and streamline administrative tasks.
2. General description	Primary functionalities and features include: <ul style="list-style-type: none">catalog management - organization of books, articles, magazines, etc.user management - maintaining a database of users.
3. Functional Requirements	<ul style="list-style-type: none">a) User registration: registration of library users to the LMS.b) Catalog search: maintaining a search system where users can search by title, author, genre or a book ID.c) Borrowing and Returning: borrowing and returning through the system.d) Renewal of items: renewing of borrowed items if there are no holds on the user.e) Reporting: administrators can generate reports on inventory, user activity and overdue items.

4.	<u>Interface Requirements</u>
a)	User Interface: a web application to interface interact with the users for the LMS
b)	Database Interface: usage of a database technology to maintain a system of user books, etc.
5.	<u>Performance Requirements</u>
a)	Response Time: should be less than 3 seconds
b)	concurrent users: LMS should be able to handle multiple simultaneous users (500)
c)	Transaction error rate: should be less than 0.5%
6.	<u>Design Constraints</u>
a)	Regulatory Compliance: data protection regulations
b)	Technology stack (Python, Django)

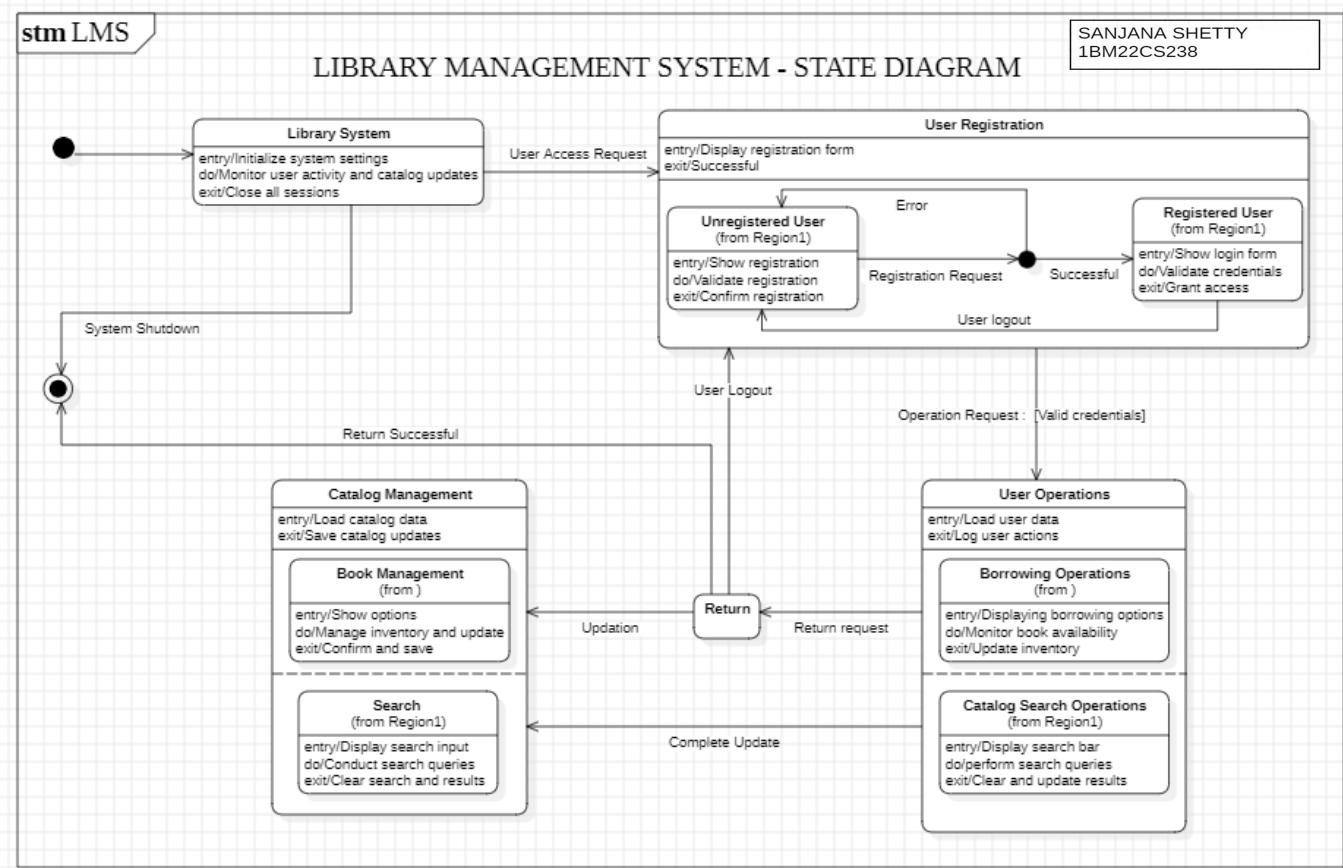
7.	<u>Non Functional Attributes</u>
a)	Security: user data must be secure and handled carefully.
b)	Reliability: reliable web app to handle multiple users.
c)	Portability: compatibility with multiple configurations.
8.	<u>Preliminary Schedule and Budget</u>
	Estimated time: 6-7 months
	Estimated total budget: 15,00,000

CLASS DIAGRAM



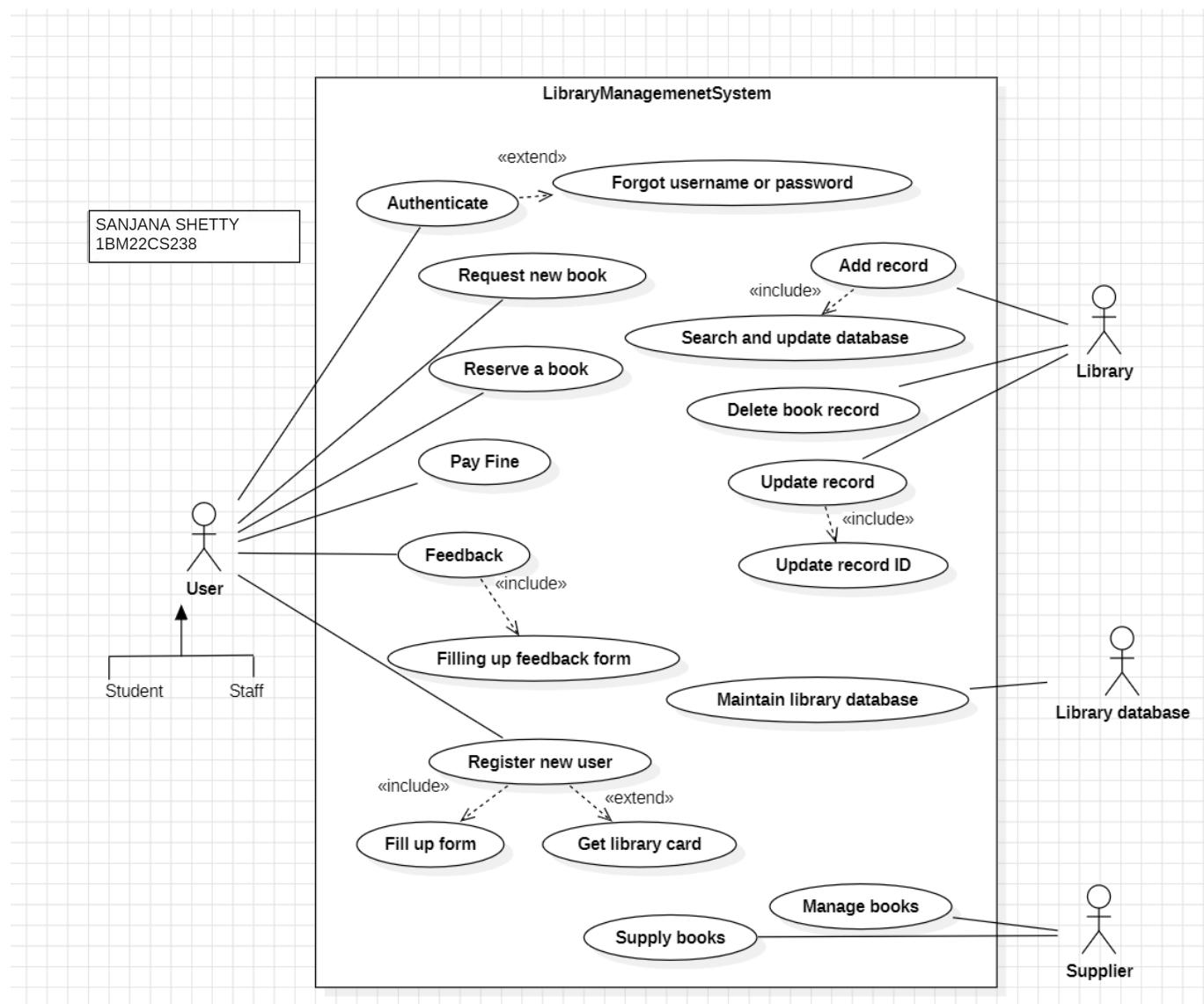
This UML class diagram illustrates the core components of a library management system. It depicts entities such as Library, Book, Member, Librarian, and their relationships. Key features include book reservations, borrowing transactions, and fine management. The diagram also includes enumerations for different member types, shifts, and statuses.

STATE DIAGRAM



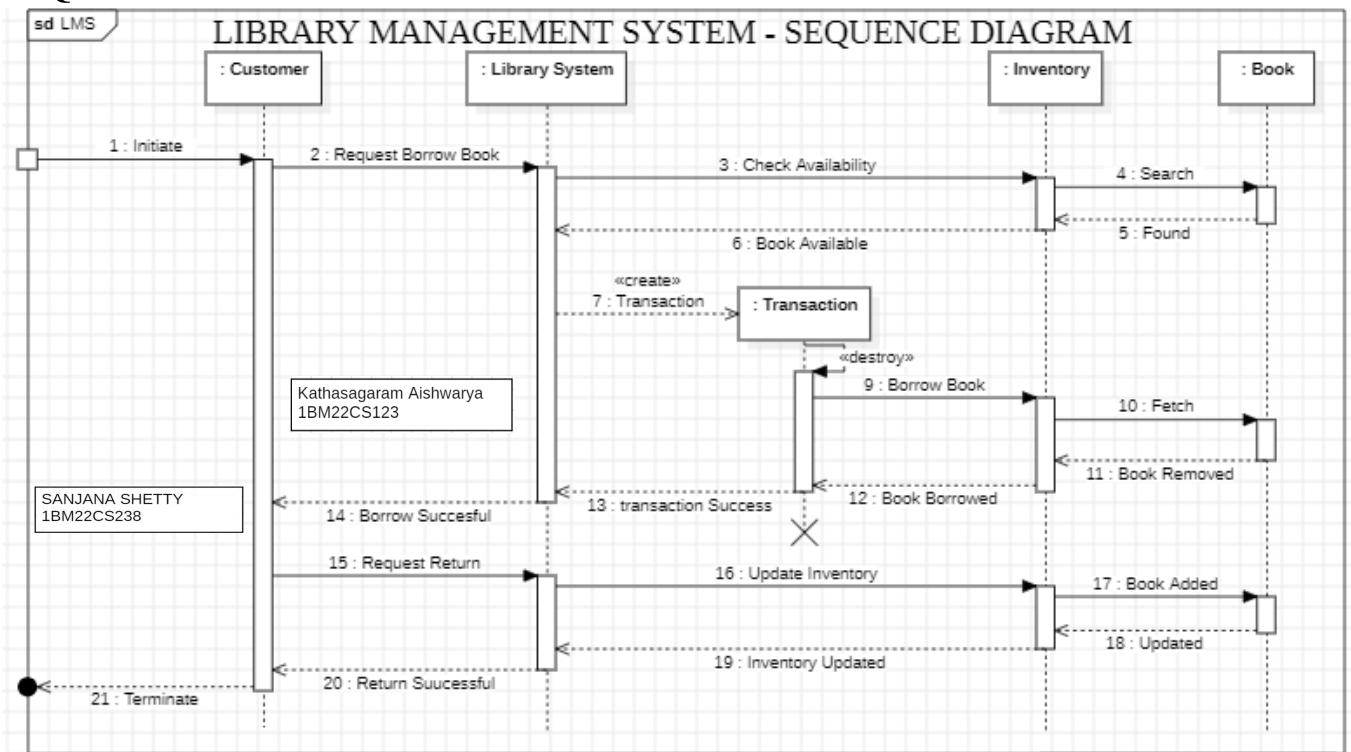
This state diagram illustrates the workflow of a library management system. It starts with the system initializing settings and monitoring user activity. Users can register and log in to access operations like catalog management, book management, borrowing operations, and catalog search. The system handles various states, including user registration, login, catalog loading, book management, search operations, and user operations. The diagram also includes error handling for unsuccessful registration or login attempts.

USE CASE DIAGRAM



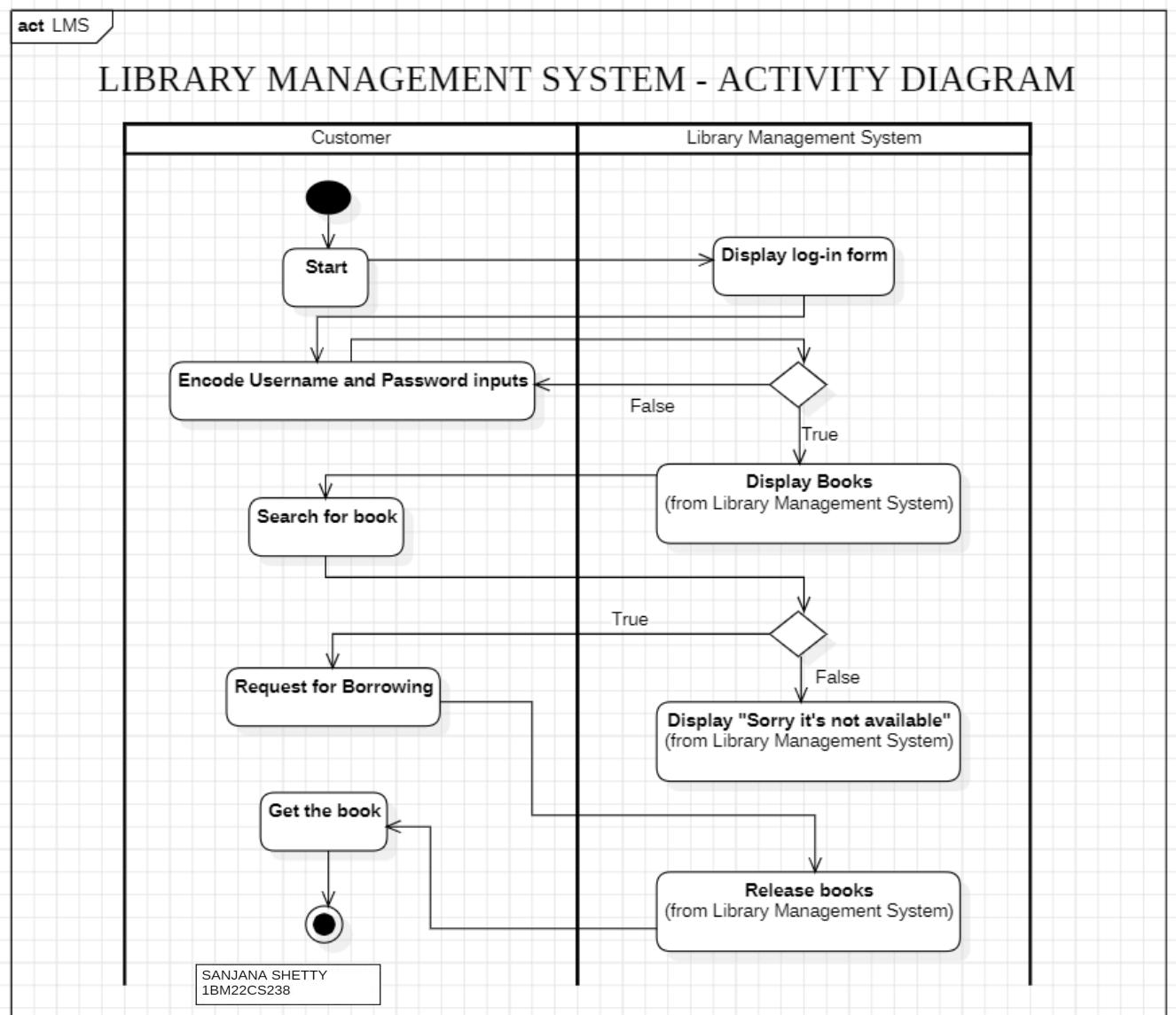
This UML use case diagram depicts the functionalities of a Library Management System. It shows how different actors, such as Users (Students and Staff), Library, Library Database, and Suppliers, interact with the system. Key use cases include Authentication, Book-related operations (Requesting, Reserving, Paying fines), Feedback, User Registration, and Library Management tasks like maintaining the database and managing books. The diagram also utilizes relationships like <<include>> and <<extend>> to represent dependencies between use cases.

SEQUENCE DIAGRAM



This sequence diagram illustrates the process of borrowing a book in a Library Management System. It shows the interactions between the Customer, Library System, Inventory, and Book objects. The sequence starts with the Customer initiating a request to borrow a book. The Library System checks the availability of the book in the Inventory. If available, a transaction is created, and the book is borrowed by the Customer. When the Customer returns the book, the Library System updates the Inventory. Finally, the process terminates.

ACTIVITY DIAGRAM



This activity diagram models the process of borrowing a book in a Library Management System. It starts with the customer logging in to the system. After successful login, the customer searches for a book. If the book is available, the customer requests to borrow it. Upon approval, the customer receives the book. When the customer is done, they return the book to the library. If the book is not available during the search, the system displays a message indicating that the book is currently unavailable.

STOCK MAINTENANCE SYSTEM

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

<p>7/10/24 SRS for Stock Maintenance System (SMS)</p> <p>1. <u>Introduction</u></p> <p>1.1 <u>Aim/purpose of this Document</u></p> <p>The purpose of the document is to outline functional and non-functional requirements of SMS. The document will be a tool to have a shared understanding of the system's objectives and functionalities.</p> <p>1.2 <u>Scope of this document</u></p> <p>To outline the expected values that will be provided to the users by the system. It includes development costs and timelines.</p> <p>1.3 <u>Overview</u></p> <p>The Stock Maintenance System aims to automate the tracking and maintenance of inventory levels and orders & stock movements.</p> <p>2. <u>General Description</u></p> <p>The primary functions of the SMS include Inventory tracking (Stock levels, locations management & maintenance) Order Management (maintenance of purchase & sales orders)</p>	<p>3. <u>Functional Requirements</u></p> <ul style="list-style-type: none">a) user registration: registering users to the stock inventoryb) Inventory management: add, delete and modify stock and product details.c) Dashboard to report stock levelsd) Analysis reports on the stock levels.e) users can create and purchase sales orders & link them to inventory items. <p>4. <u>Interface Requirements</u></p> <ul style="list-style-type: none">a) User interface: a web app to interact with usersb) Database interface: usage of a database interface, with:<ul style="list-style-type: none">a) real-time data communication between the system & the database ensuring constant stock level updatesb) shared data streams such as sales orders, purchase orders, and stock information
---	--

Date _____
Page _____

5. Performance Requirements:

- System should be able to handle 500 concurrent users without any performance issues.
- Processing of stock updates & reflecting changes should happen in less than 2 seconds.
- Maximum error rate of stock transaction = 0.1%.

6. Design Constraints:

- The system should be compatible with various databases.
- Should be developed with scalable architecture to accommodate for increasing data volume.
- System should adhere to data privacy regulations.

7. Non-functional attributes:

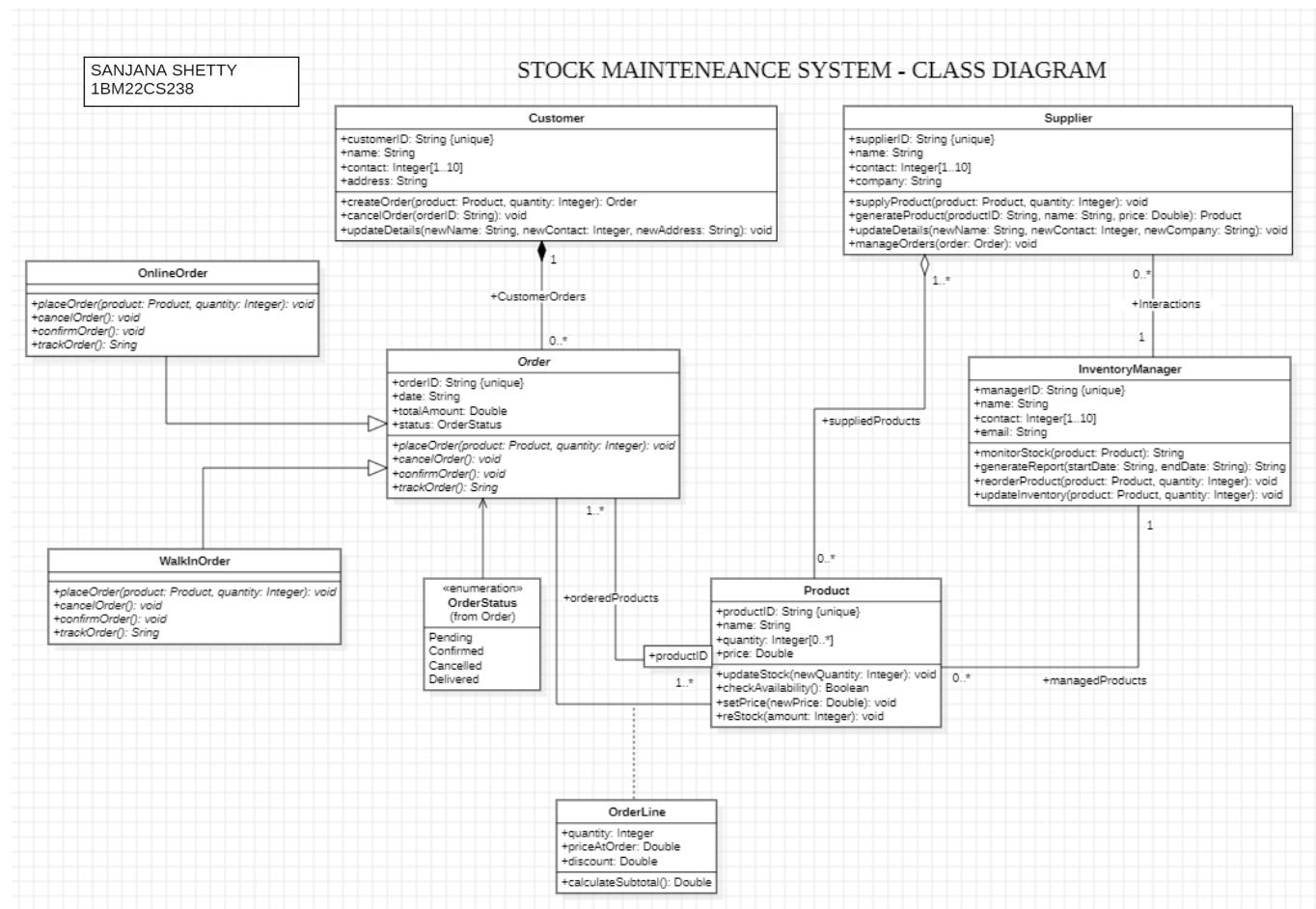
- Security : ensuring only authorized users can view or modify stock data.
- Profitability : system should be deployable on all platforms.
- Scalability : The system should be able to handle increased load.

6. Preliminary Schedule and Budget

Schedule: 6 months for design, developing, testing & deployment.

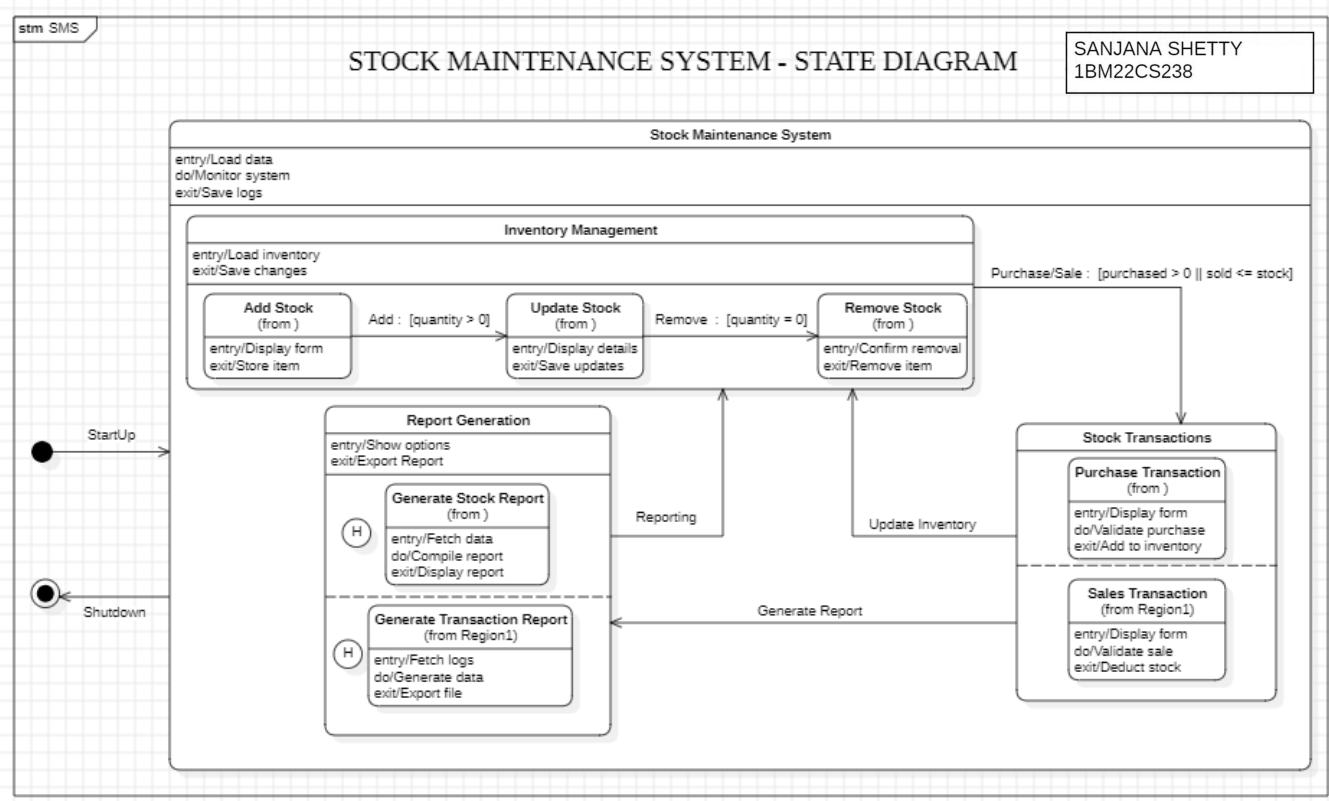
Budget : \$50,000 (hardware, software licenses, development time, & maintenance for the first year)

CLASS DIAGRAM



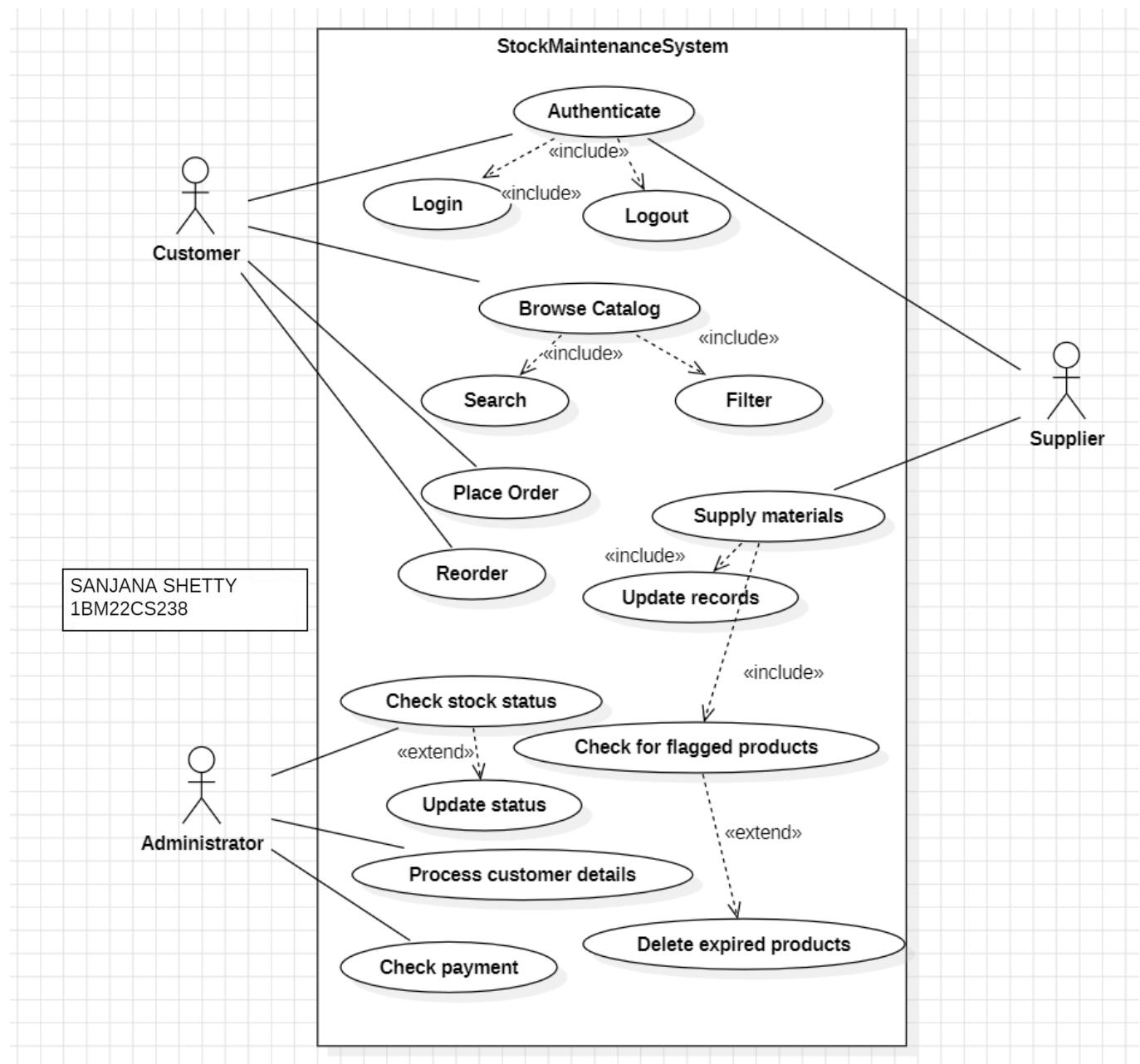
This UML class diagram illustrates the structure of a Stock Maintenance System. It depicts various entities such as Customer, Supplier, Inventory Manager, Product, and Order, along with their attributes and relationships. Key relationships include: Customers placing orders, Suppliers supplying products, Inventory Manager managing stock, and Orders involving OrderLines. The diagram also includes enumerations for OrderStatus, representing the different stages of an order. This diagram provides a high-level overview of the system's components and their interactions.

STATE DIAGRAM



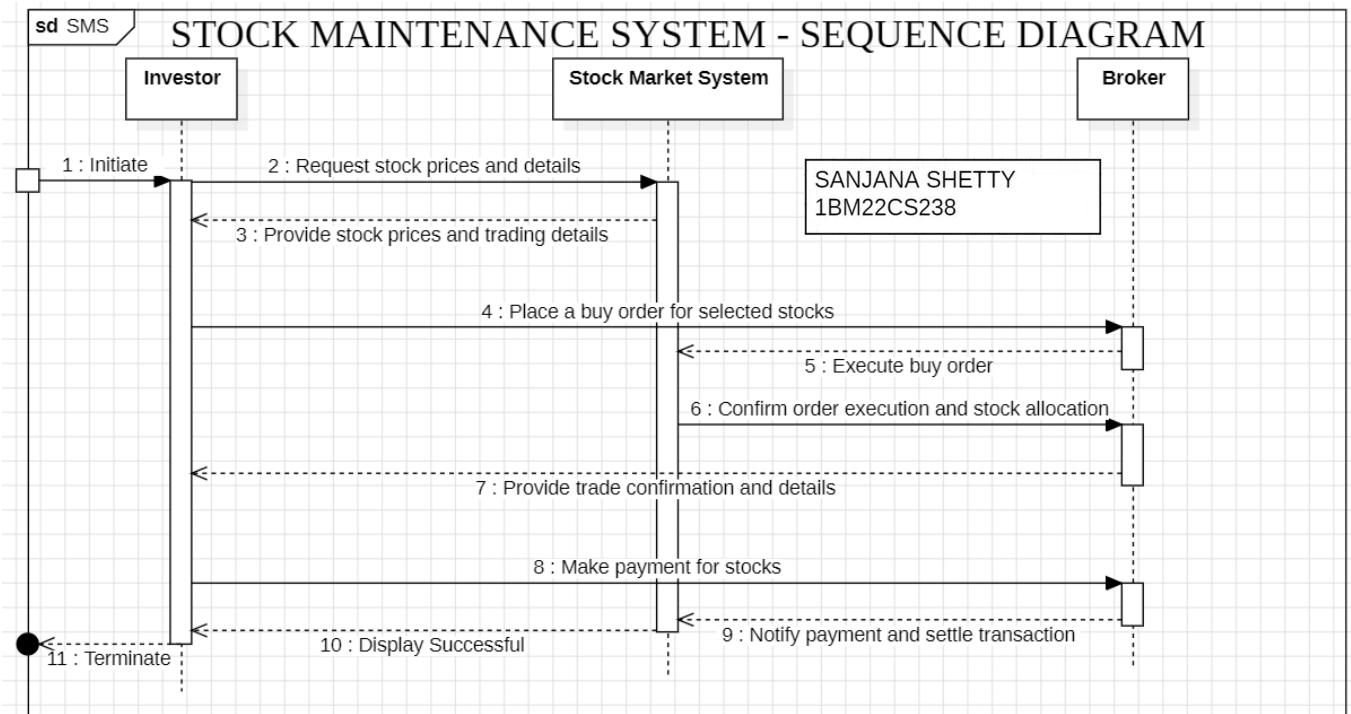
This state diagram illustrates the workflow of a Stock Maintenance System. It starts with the system loading data and monitoring operations. The system can then enter Inventory Management mode, where actions like adding stock, updating stock details, and removing stock can be performed. The system also supports Report Generation, including generating stock reports and transaction reports. The system can transition between these modes and ultimately shut down, saving logs before exiting.

USECASE DIAGRAM

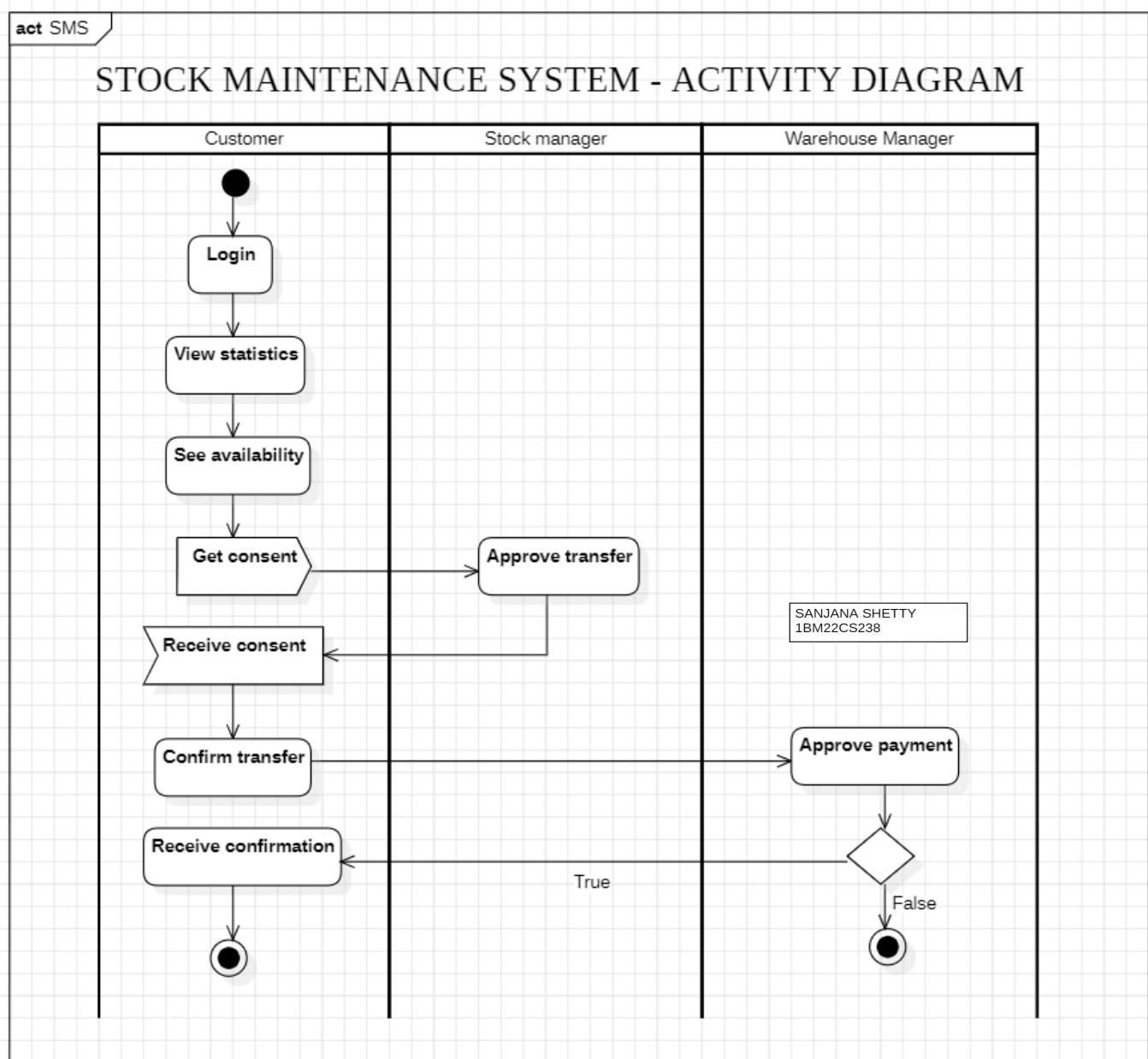


This UML use case diagram models the functionalities of a Stock Maintenance System. It shows how various actors, such as Customers, Suppliers, and Administrators, interact with the system. Key use cases include Authentication (Login and Logout), browsing and searching the catalog, placing orders, reordering, supplying materials, checking stock status, updating records, and managing flagged products. The diagram also utilizes relationships like <<include>> and <<extend>> to represent dependencies and variations between use cases.

SEQUENCE DIAGRAM



ACTIVITY DIAGRAM



This activity diagram outlines the process of transferring goods within a stock maintenance system. It involves the customer initiating the transfer request, obtaining necessary approvals from the stock manager and warehouse manager, and finally receiving confirmation upon successful transfer.

PASSPORT AUTOMATION SYSTEM

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

<p>7/10/24 SRS for Passport Automation System (PAS)</p> <p>1. <u>Introduction</u></p> <p>1.1 Purpose of this Document:</p> <p>The purpose of the document is to outline the requirements, design, and functionality of the Passport Automation System. This system is intended to automate and streamline the process of passport application, verification, and issuance, reducing manual errors and speeding up the overall process.</p> <p>1.2 Scope of this Document:</p> <p>This document details the functionalities of the PAS, which includes the submission of applications & documents, appointments, fees, application status. The document also covers timeline & cost estimates.</p> <p>1.3 Overview:</p> <p>The PAS allows citizens to apply for passports online, track their application status, & receive notifications at various stages. It facilitates verification & authentication for passport officials.</p>	<p>2. General Description</p> <p>The PAS aims at simplifying process of applying for & receiving passports. It caters to various users like applicants, passport officials, & administrators. Key features include online application portal, document uploads, appointment scheduling, payments, and real-time tracking of the application process. The system's objective is to improve user experience, etc.</p> <p>3. Functional Requirements:</p> <ol style="list-style-type: none">User registration & login for submitting passport applications.Personal details (e.g. name, address, nationality, etc.) Required documents (e.g. proof of identity, proof of residence, etc.)Support appointment scheduling for document verification and biometric data collection at regional passport offices.It should generate & send email or SMS notification to users regarding their application status.System should include a payment gateway.Reviewed applications by passport officials can be marked as approved, rejected or under review.Every user is associated with a unique ID.
--	--

2. General Description

The PAS aims at simplifying process of applying for & receiving passports. It caters to various users like applicants, passport officials, & administrators. Key features include online application portal, document uploads, appointment scheduling, payments, and real-time tracking of the application process. The system's objective is to improve user experience, etc.

3. Functional Requirements:

- a) User registration & login for submitting passport applications.
- b) Personal details (e.g. name, address, nationality, etc.)
Required documents (e.g. proof of identity, proof of residence, etc.)
- c) Support appointment scheduling for document verification and biometric data collection at regional passport offices.
- d) It should generate & send email or SMS notifications to users regarding their application status.
- e) System should include a payment gateway.
- f) Reviewed applications by passport officials can be marked as approved, rejected or under review.
- g) Every user is associated with a unique ID.

7. Non-Functional Attributes:

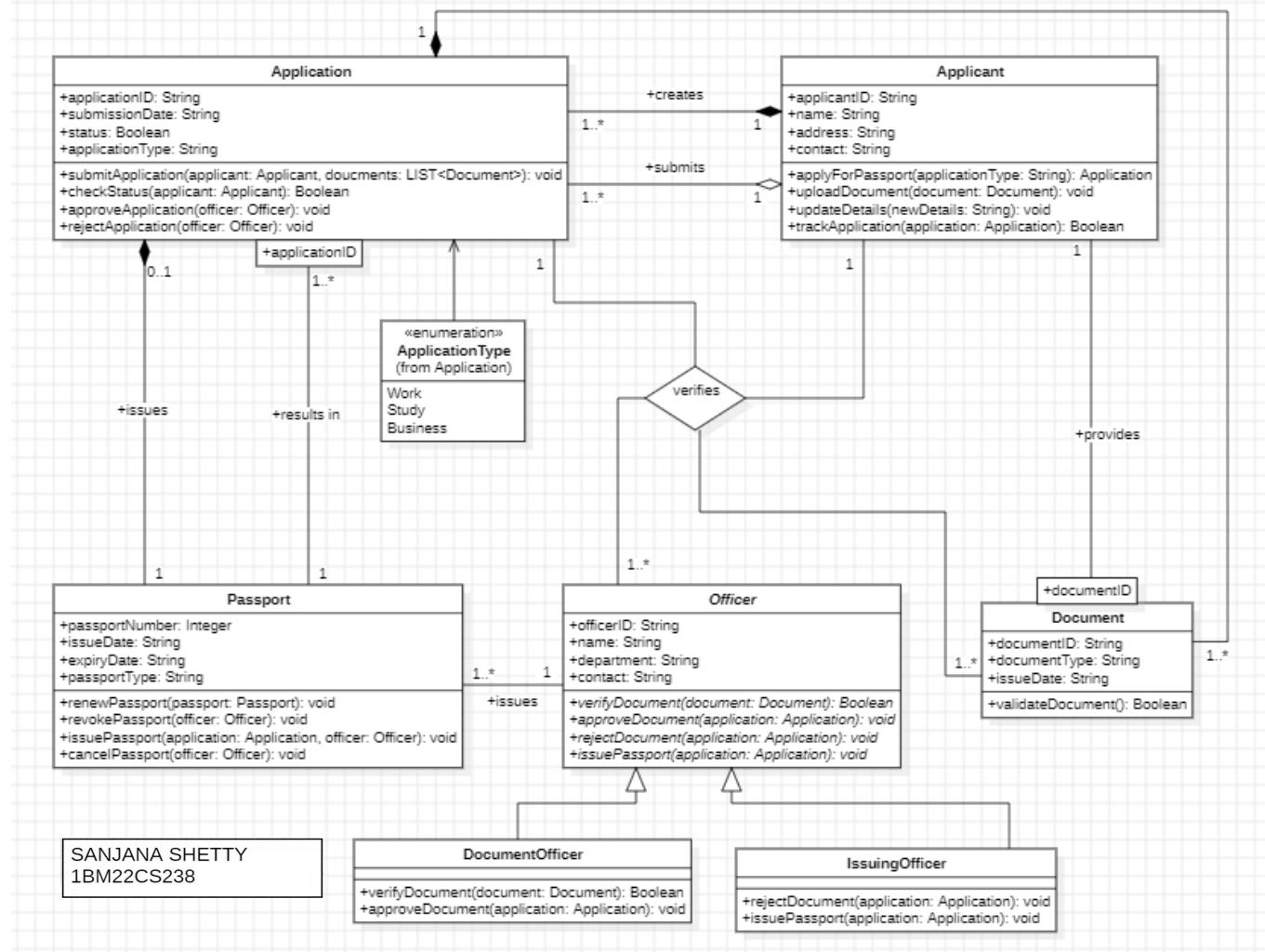
- a) Security: implement multi-factor authentication & secure data encryption for all data transfers.
- b) Portability: should be accessible on various devices.
- c) Reusability: different modules should be reusable for other government services.

8. Preliminary Schedule and Budget

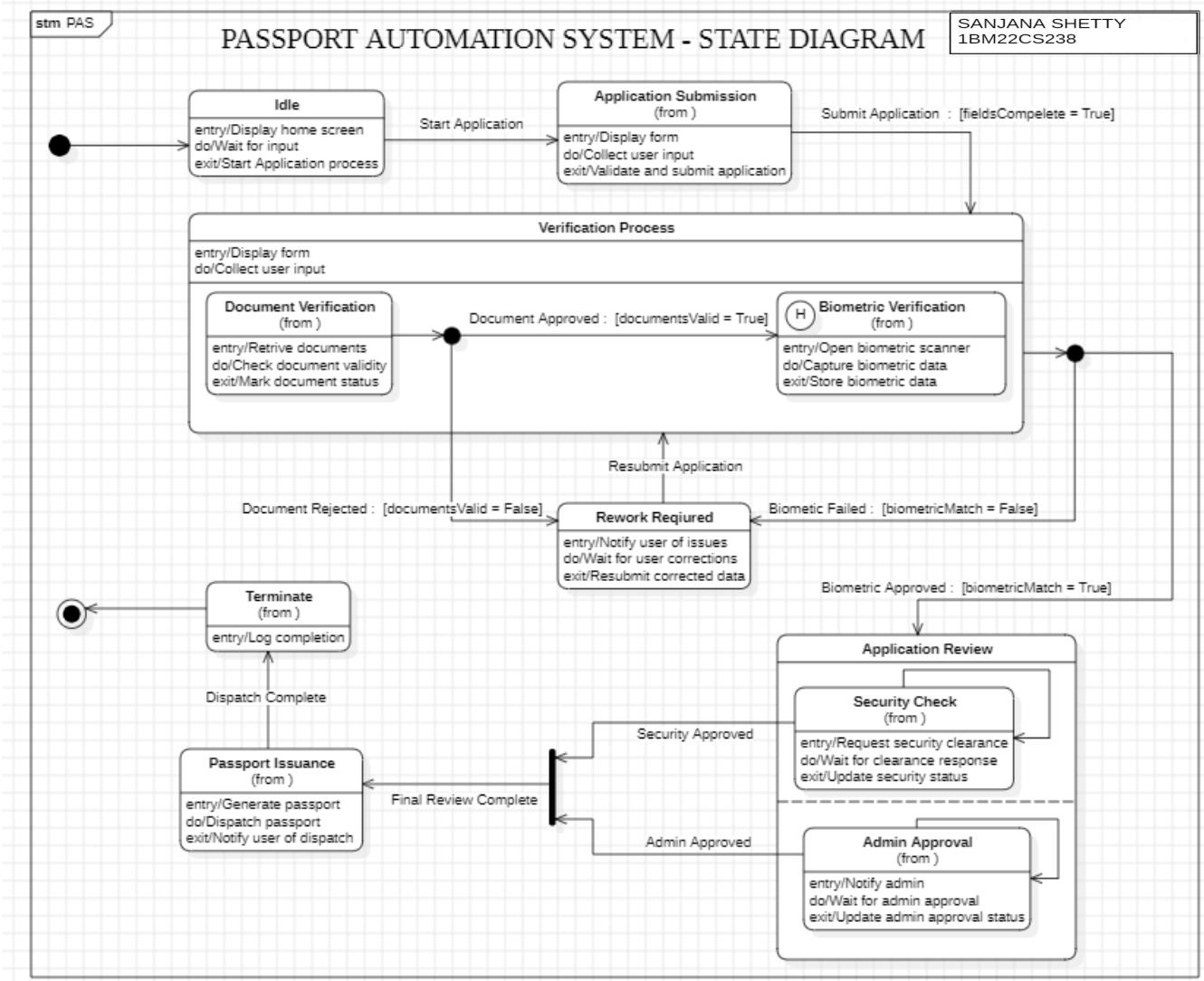
- Schedule: 9 months (design, development, testing, deployment)
- Budget: \$ 75,000

CLASS DIAGRAM

PASSPORT AUTOMATION SYSTEM - CLASS DIAGRAM

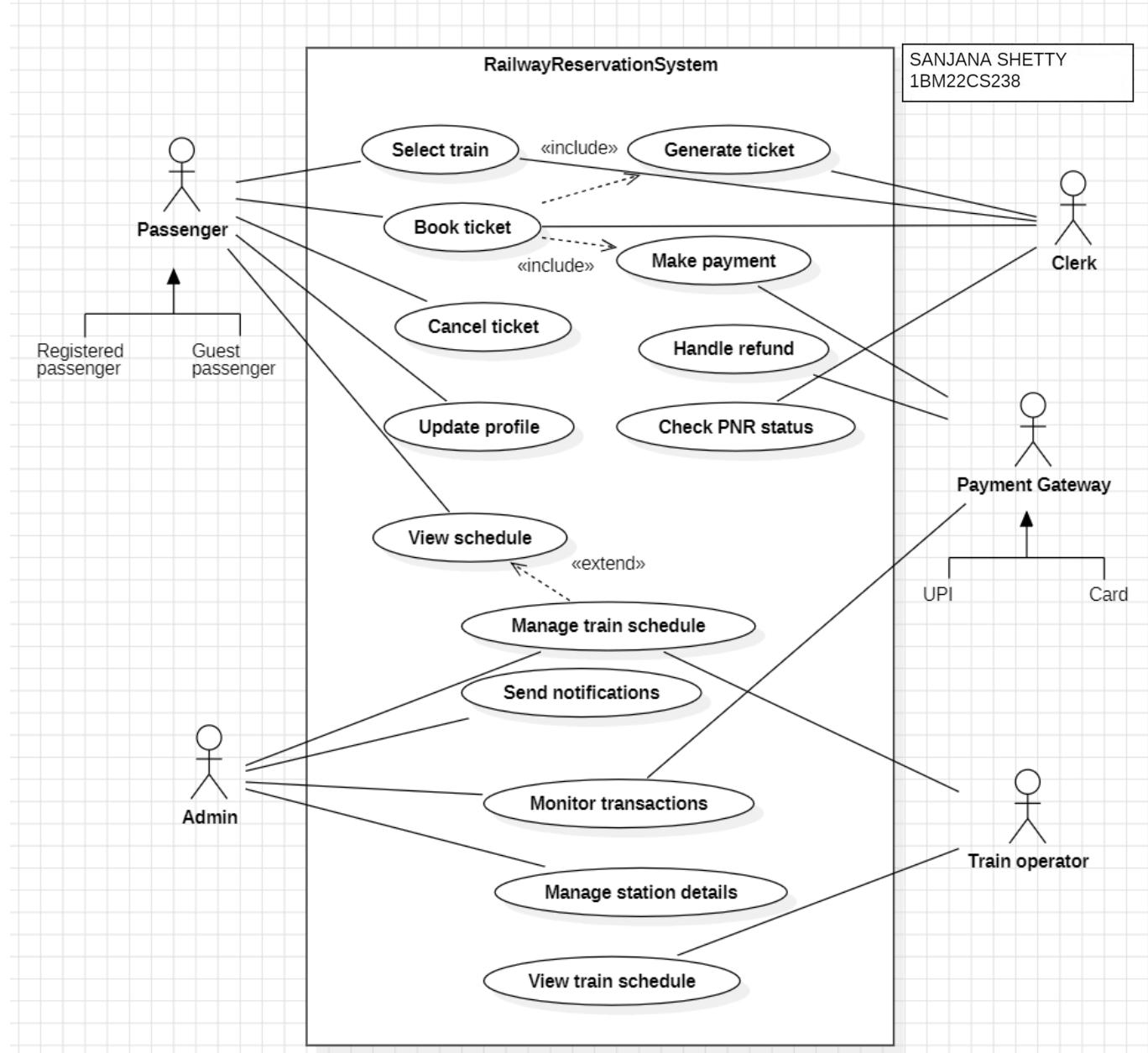


STATE DIAGRAM



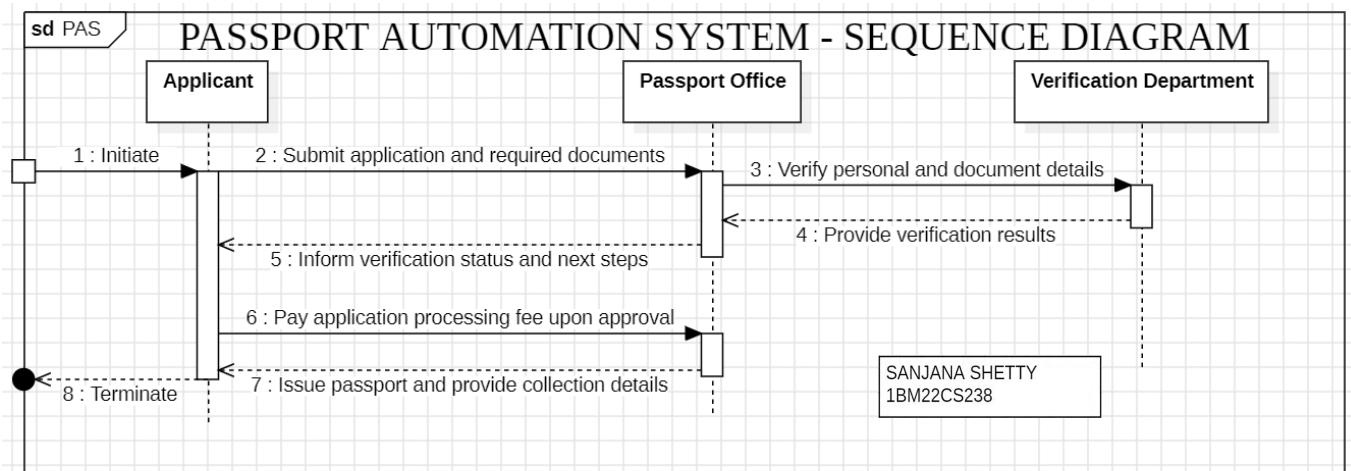
This state diagram illustrates the workflow of a Passport Automation System. It starts in an Idle state, displaying a home screen. The system transitions to the Application Submission state where the user fills out the application form. The system then verifies the application's completeness. If complete, it moves to the Verification Process, where documents are verified and biometric data is captured. Based on the verification results, the system may reject the application, require rework, or proceed to the Application Review stage. In the review stage, security checks and administrative approval are obtained. Finally, if approved, the passport is issued and the application is marked as complete.

USE CASE DIAGRAM



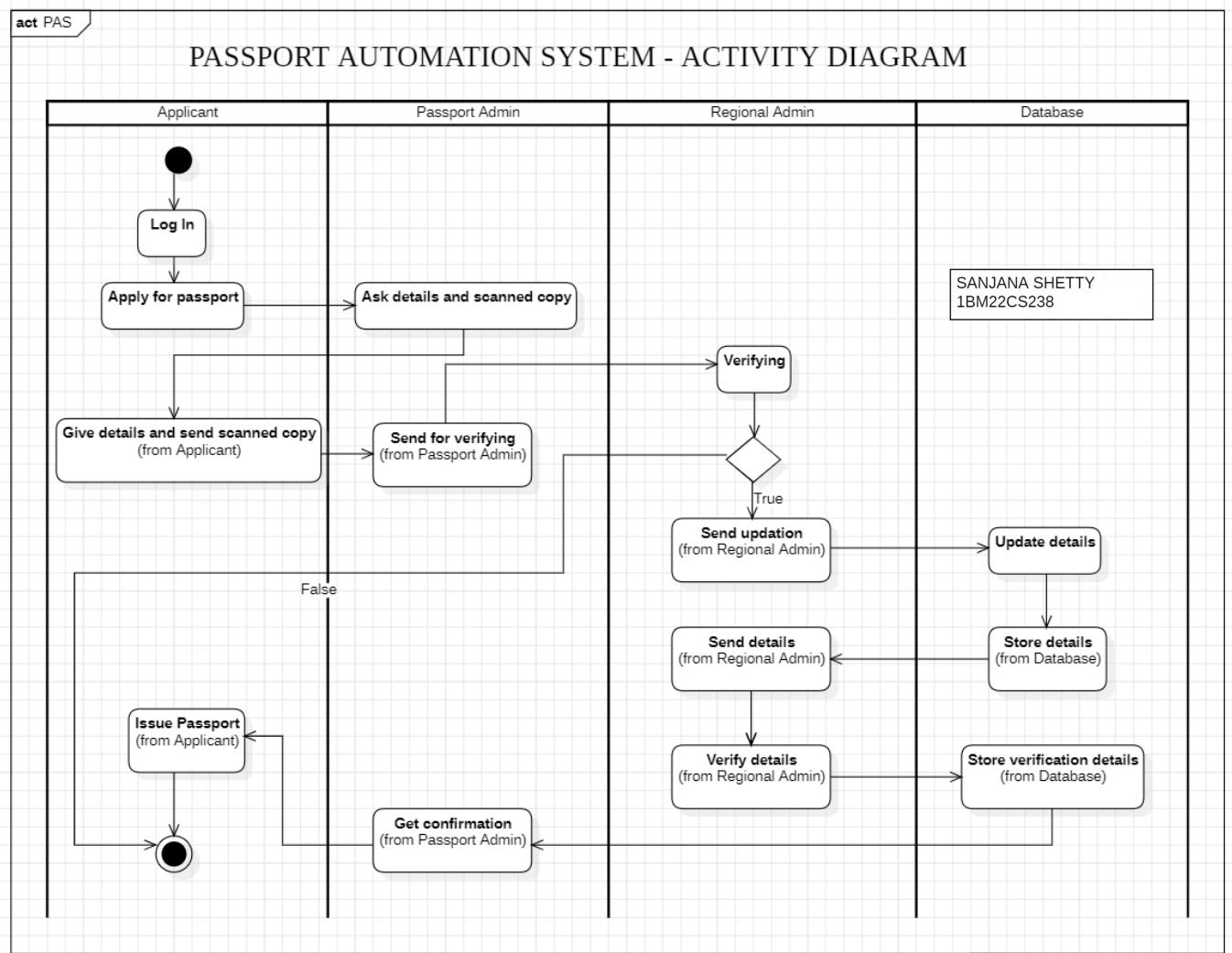
This UML use case diagram models the functionalities of a Railway Reservation System. It shows how different actors, such as Passengers (Registered and Guest), Clerk, Admin, Payment Gateway, Train Operator, and Station, interact with the system. Key use cases include booking tickets, canceling tickets, making payments, checking PNR status, viewing schedules, managing train schedules, and monitoring transactions. The diagram also utilizes relationships like <<include>> and <<extend>> to represent dependencies and variations between use cases.

SEQUENCE DIAGRAM



This sequence diagram illustrates the process of a passport application in a Passport Automation System. It shows the interactions between the Applicant, Passport Office, and Verification Department. The sequence begins with the Applicant initiating the process by submitting the application and required documents. The Verification Department verifies the details and informs the Passport Office of the results. Upon approval, the Applicant pays the processing fee. Finally, the Passport Office issues the passport and provides collection details, concluding the process.

ACTIVITY DIAGRAM



This activity diagram illustrates the workflow of a Passport Application process in a Passport Automation System. It starts with the Applicant logging in and applying for a passport. The Passport Admin then asks for details and scanned copies of documents. The Applicant provides the details and sends the scanned copies. The Passport Admin sends the details to the Regional Admin for verification. If the Regional Admin verifies the details, they are updated in the database. If not verified, the Regional Admin sends the details back to the Passport Admin, who then verifies them and stores the verification details in the database. Finally, the Applicant issues the passport and receives confirmation from the Passport Admin.