VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED MODELING

Submitted by

Shuvam Rajbanshi
(1BM22CS275)

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 **Shuvam Rajbanshi (1BM22CS275),** 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.

Radhika A D
Assistant Professor
Department of CSE
BMSCE, Bengaluru

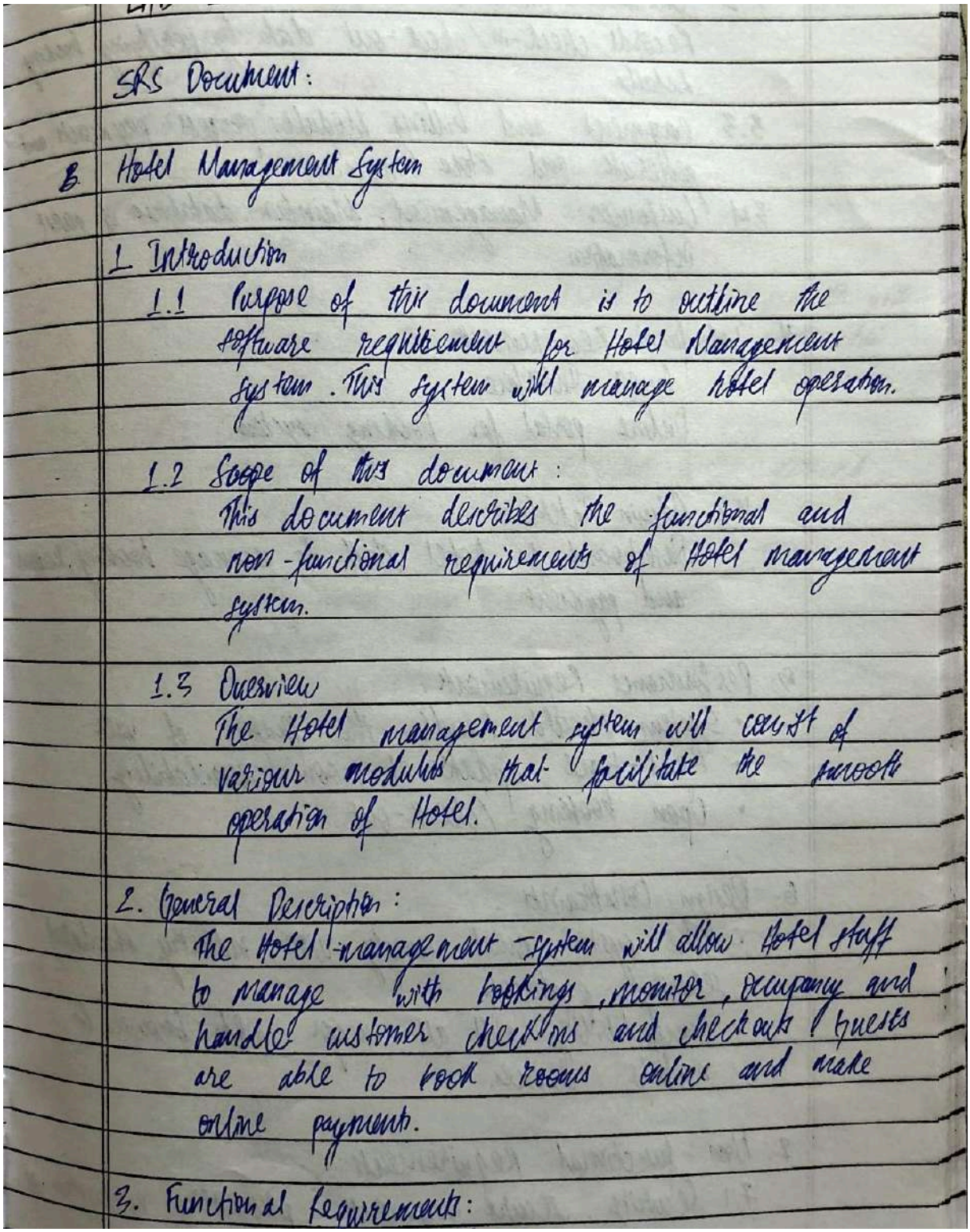
Dr. Kavitha Sooda
Professor and Head
Department of CSE
BMSCE, Bengaluru

Table of Content

https://github.com/Shuvam00/OOMD_1BM22CS275

Hotel Management System

Software Requirements Specification (SRS)



SRS Document:

8. Hotel Management System

1. Introduction

1.1 Purpose of this document is to outline the software requirement for Hotel Management system. This system will manage hotel operation.

1.2 Scope of this document:
This document describes the functional and non-functional requirements of Hotel management system.

1.3 Overview
The Hotel management system will consist of various modules that facilitate the smooth operation of Hotel.

2. General Description:
The Hotel management system will allow Hotel staff to manage with bookings, monitor occupancy and handle customer check-ins and check-outs. Guests are able to book rooms online and make online payments.

3. Functional Requirements:

details

3.3 payment and billing Module. process payments and, generate and store invoices

3.4 Customer Management. Maintain database of guest information

4. Interface Requirements

4.1 Guest Interface

Online portal for booking system

4.2 Admin Interface

Dashboard for hotel staff to manage bookings, rooms and payment

5. Performance Requirement.

- System should handle the traffic of user
- Real-time update of room availability
- Upon booking / check-out.

6. Design Constraints.

- The system must comply with industry standard security protocols
- Compatibility with all major web browser & mobile devices.

7. Non-functional Requirements.

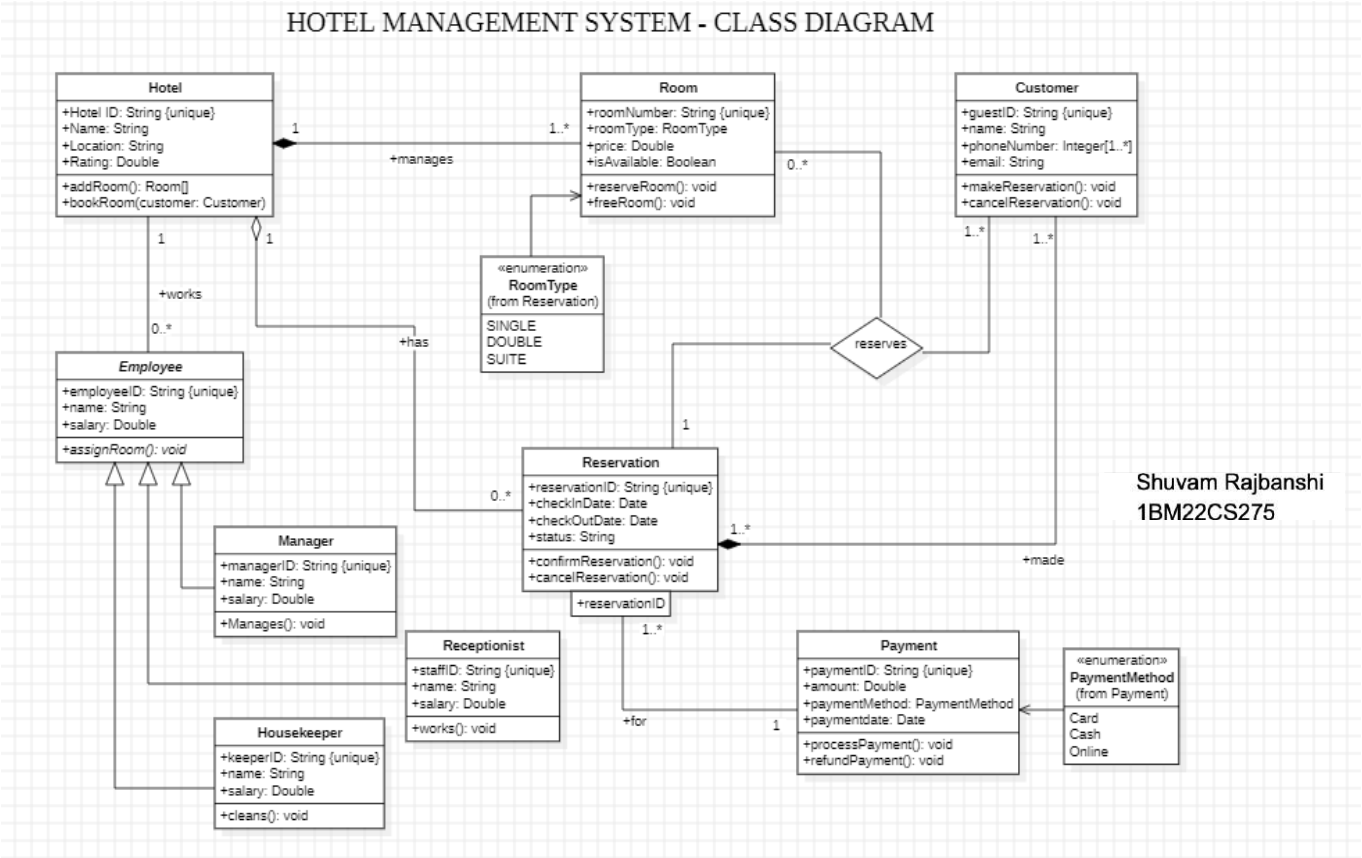
7.3 Usability
7.4 Performance
7.5 Scalability

8. Preliminary Schedule and Budget.
8.1 Development duration
- Estimated to be completed in 4 months with one month for testing and feedback integration.

8.2 Cost Estimated
- Development cost is projected to be around 4,80,000 INR.

Development Team : 60K
Hardware : 5K
Software license : 10K
Miscellaneous cost = 5K
Total Budget : 80K

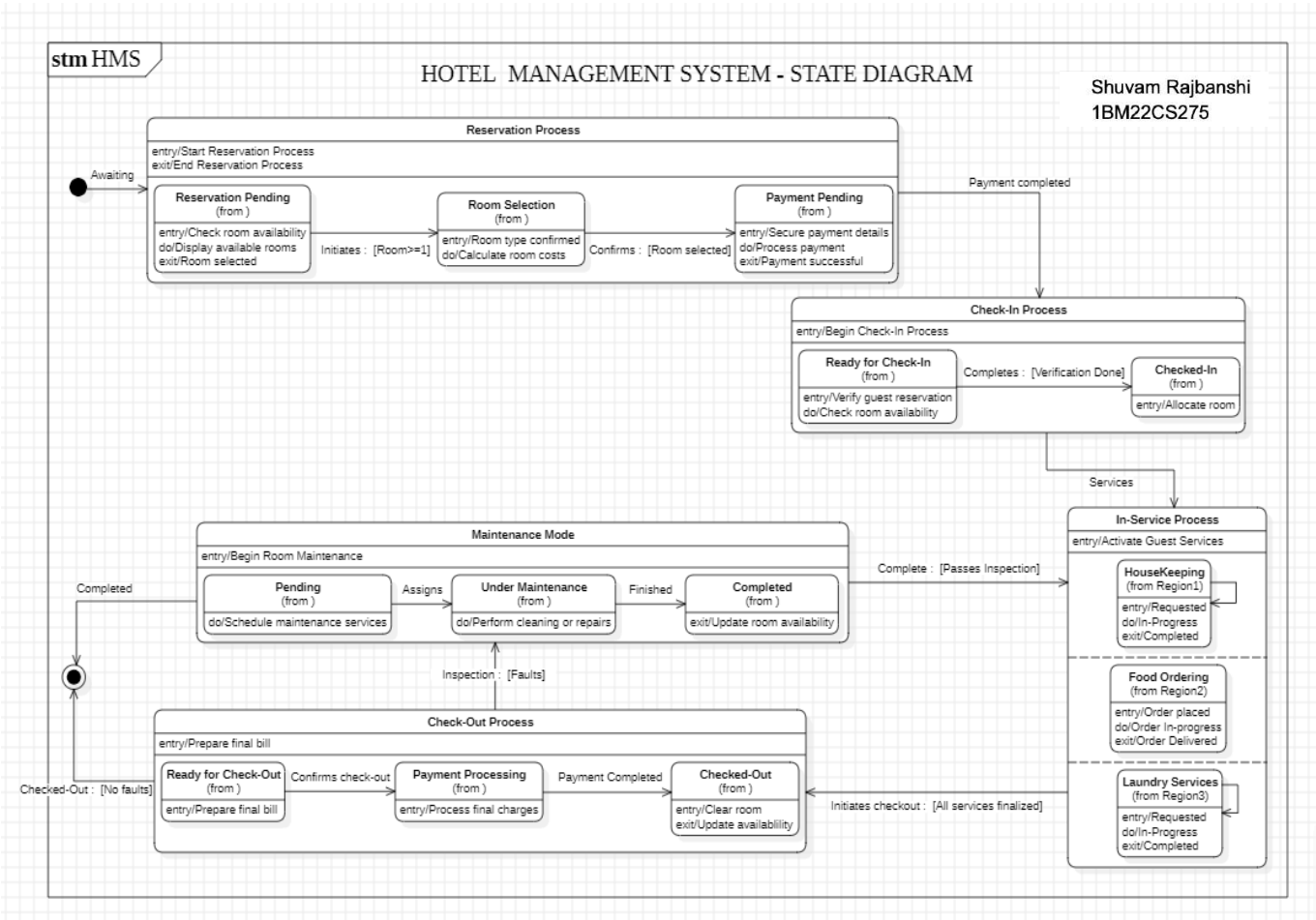
CLASS DIAGRAM



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

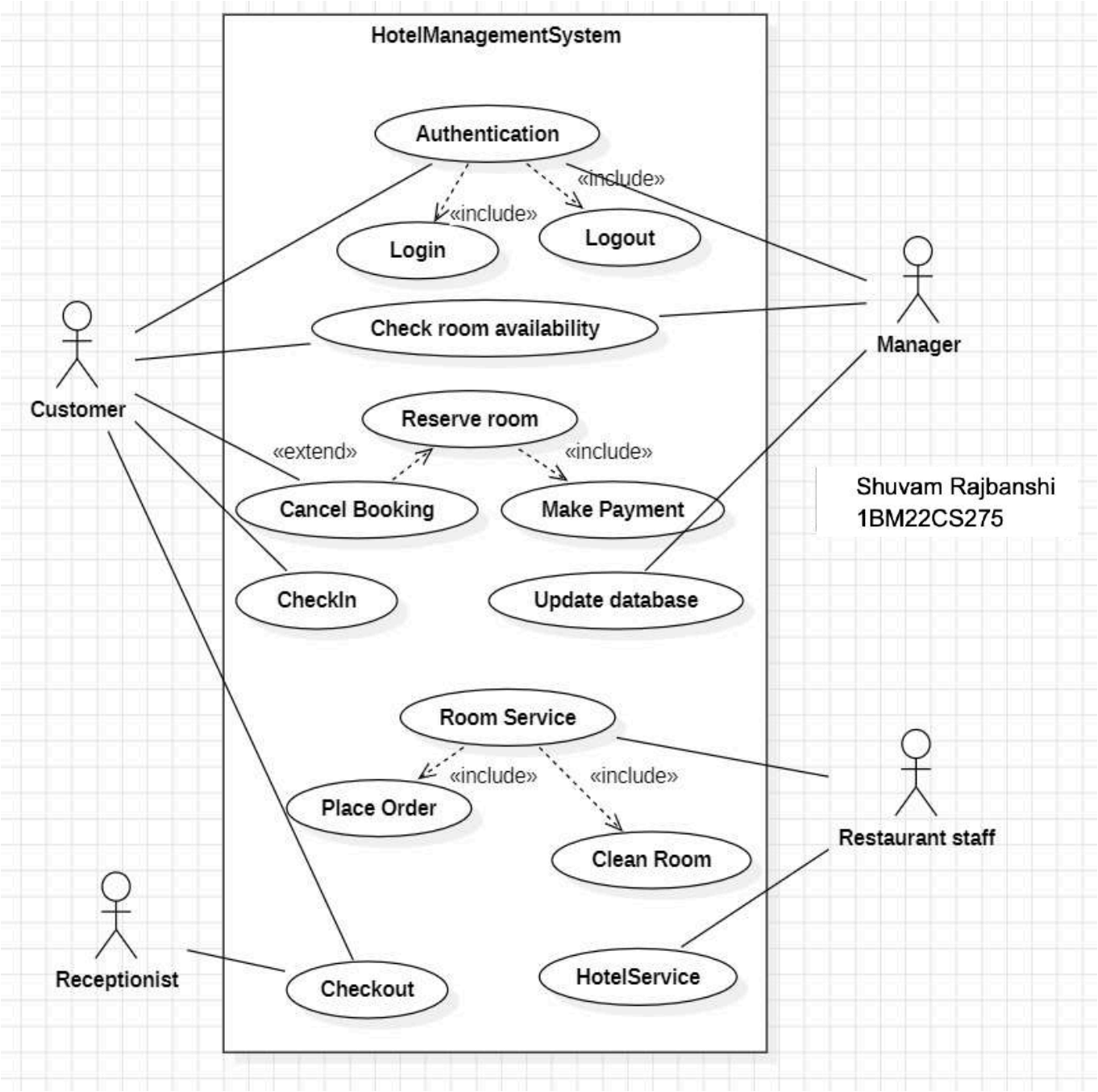
class, which records details like checkInDate, checkOutDate, and status, and is associated with Payment, handling methods such as processPayment() and refundPayment(). The system includes Employees, categorized as Managers, Receptionists, and Housekeepers, each with specific responsibilities such as 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



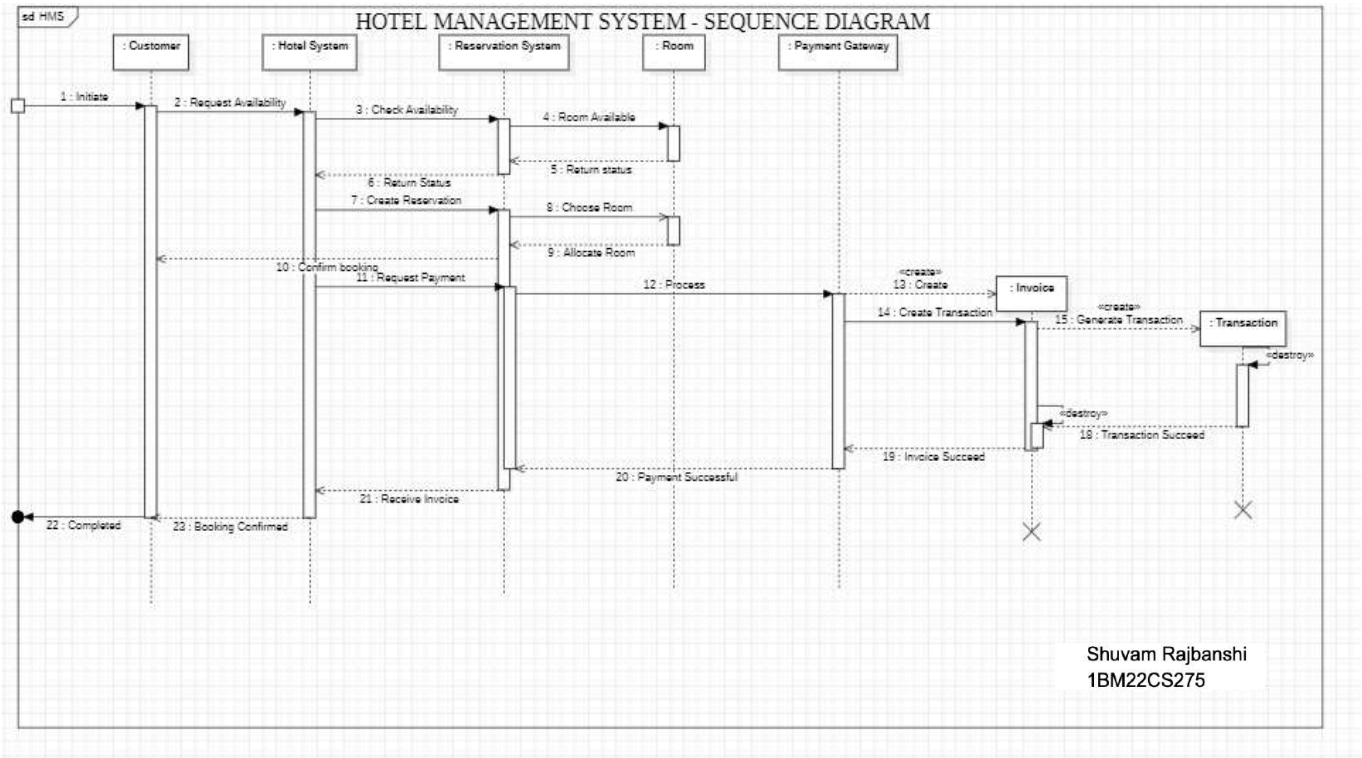
The state diagram depicts the Hotel Management System processes, starting with the Reservation Process (from Reservation Pending to payment completion), followed by the Check-in Process (Ready for Check-In to Checked-In). During the stay, the In-Service Process manages tasks like Housekeeping, Food Ordering, 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



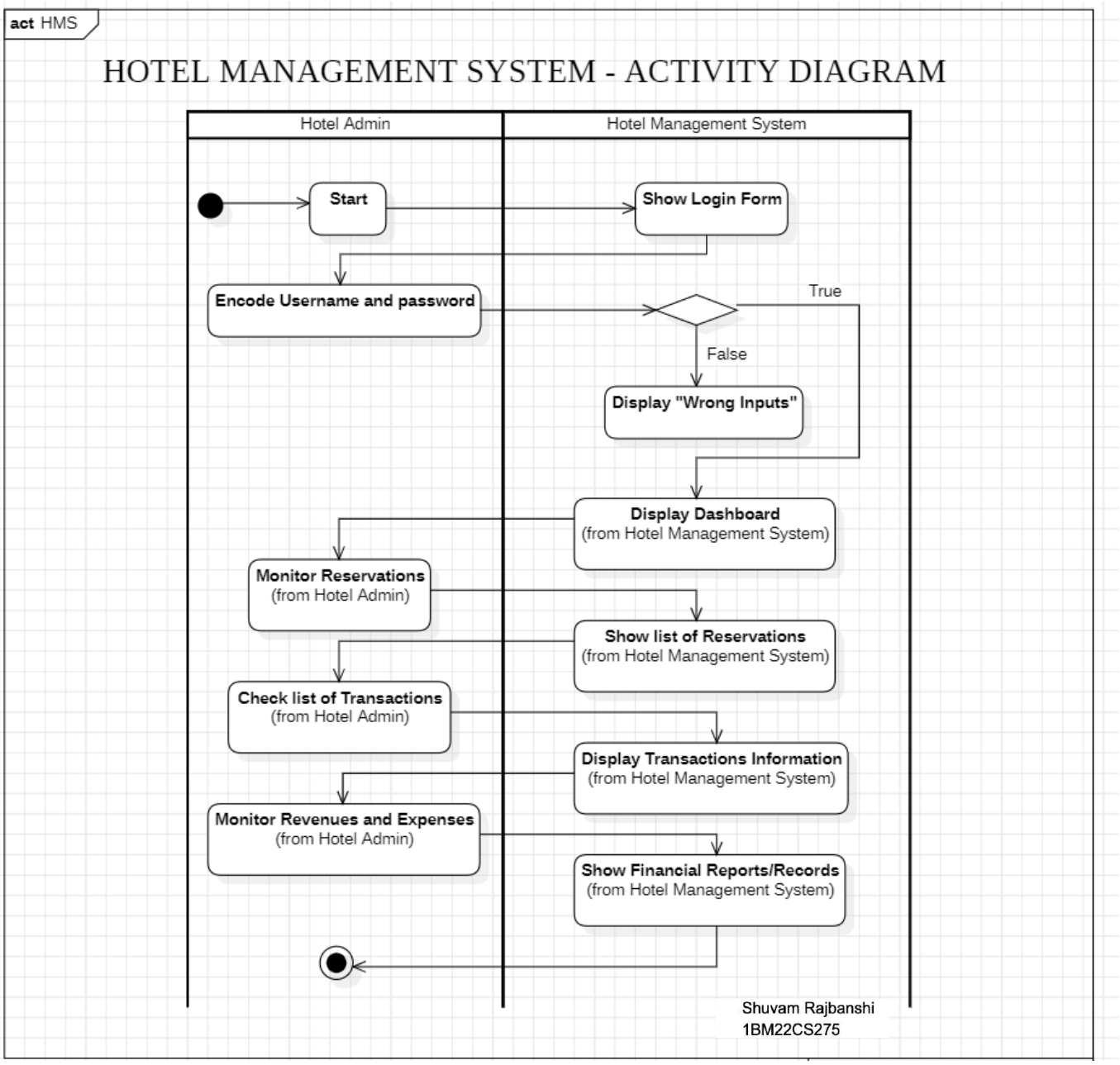
The use case diagram depicts the Hotel Management System, highlighting interactions between Customer, Manager, Receptionist, and Restaurant Staff. Key actions include Authentication, Room Reservation, Payment, Check-in, Room Service, and Checkout. It shows relationships like include and extend to represent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

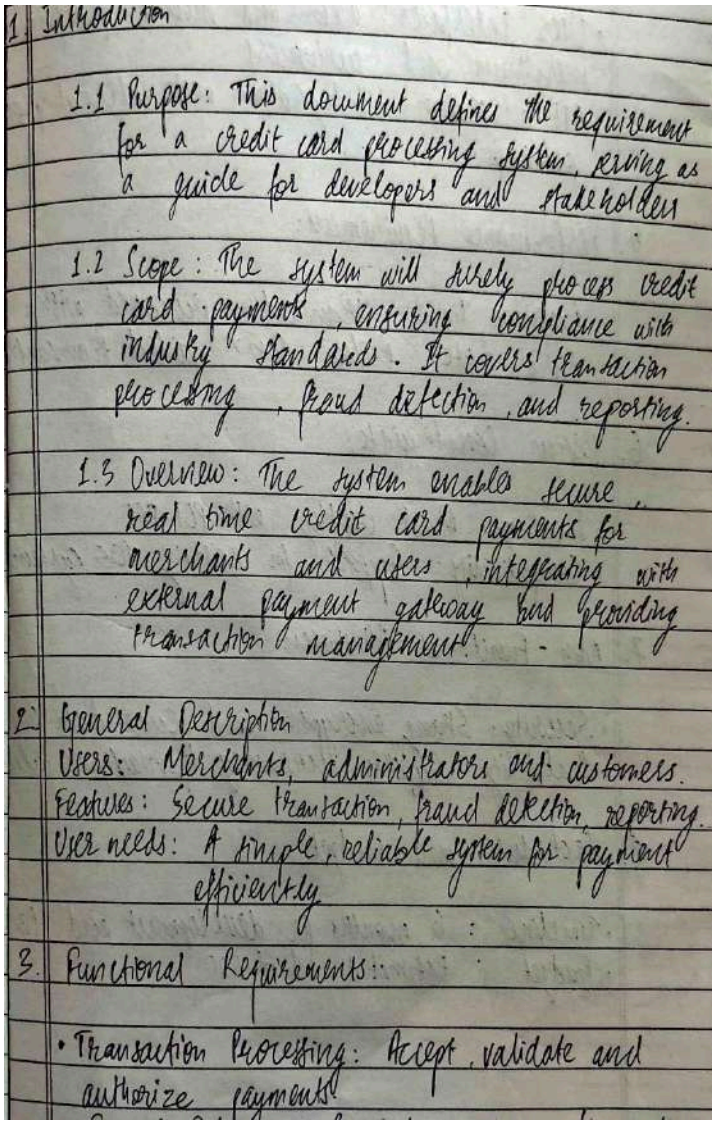
ACTIVITY DIAGRAM



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)



1. Introduction

1.1 Purpose: This document defines the requirement for a credit card processing system, serving as a guide for developers and stakeholders

1.2 Scope: The system will securely process credit card payments, ensuring compliance with industry standards. It covers transaction processing, fraud detection, and reporting.

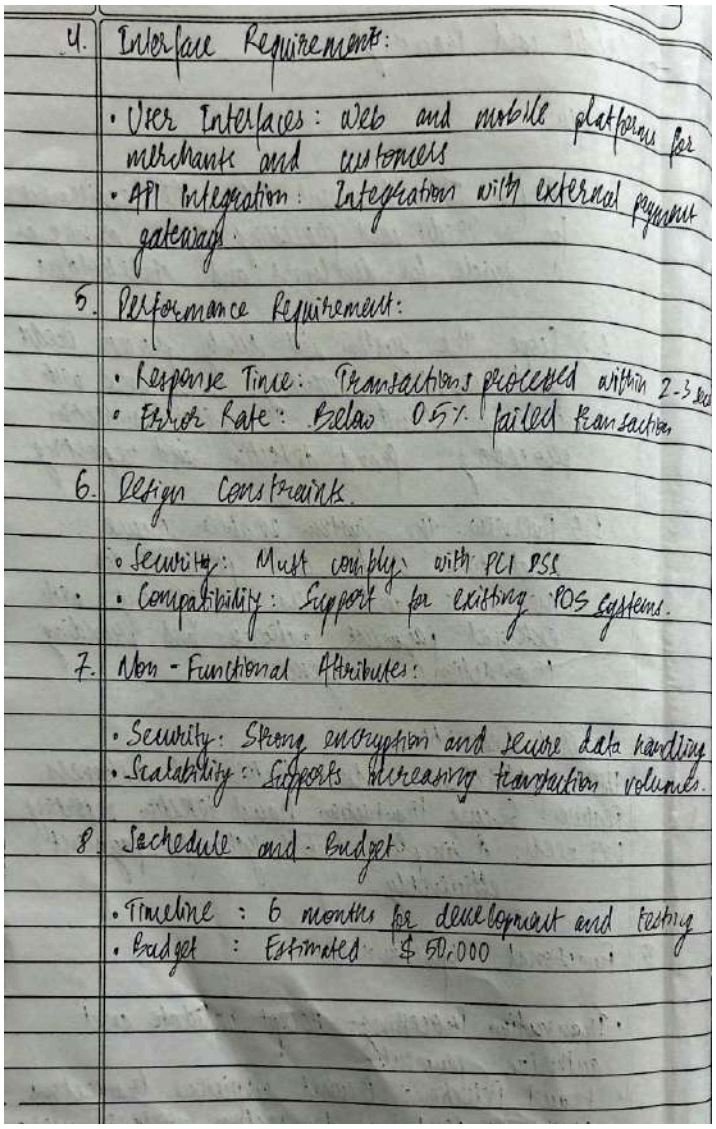
1.3 Overview: The system enables secure, real time credit card payments for merchants and users, integrating with external payment gateway and providing transaction management.

2. General Description

Users: Merchants, administrators and customers.
Features: Secure transaction, fraud detection, reporting.
User needs: A simple, reliable system for payment efficiently

3. Functional Requirements:

- Transaction Processing: Accept, validate and authorize payments



4. Interface Requirements:

- User Interfaces: Web and mobile platforms for merchants and customers
- API Integration: Integration with external payment gateways

5. Performance Requirement:

- Response Time: Transactions processed within 2-3 sec
- Error Rate: Below 0.5% failed transaction

6. Design Constraints

- Security: Must comply with PCI DSS
- Compatibility: Support for existing POS systems.

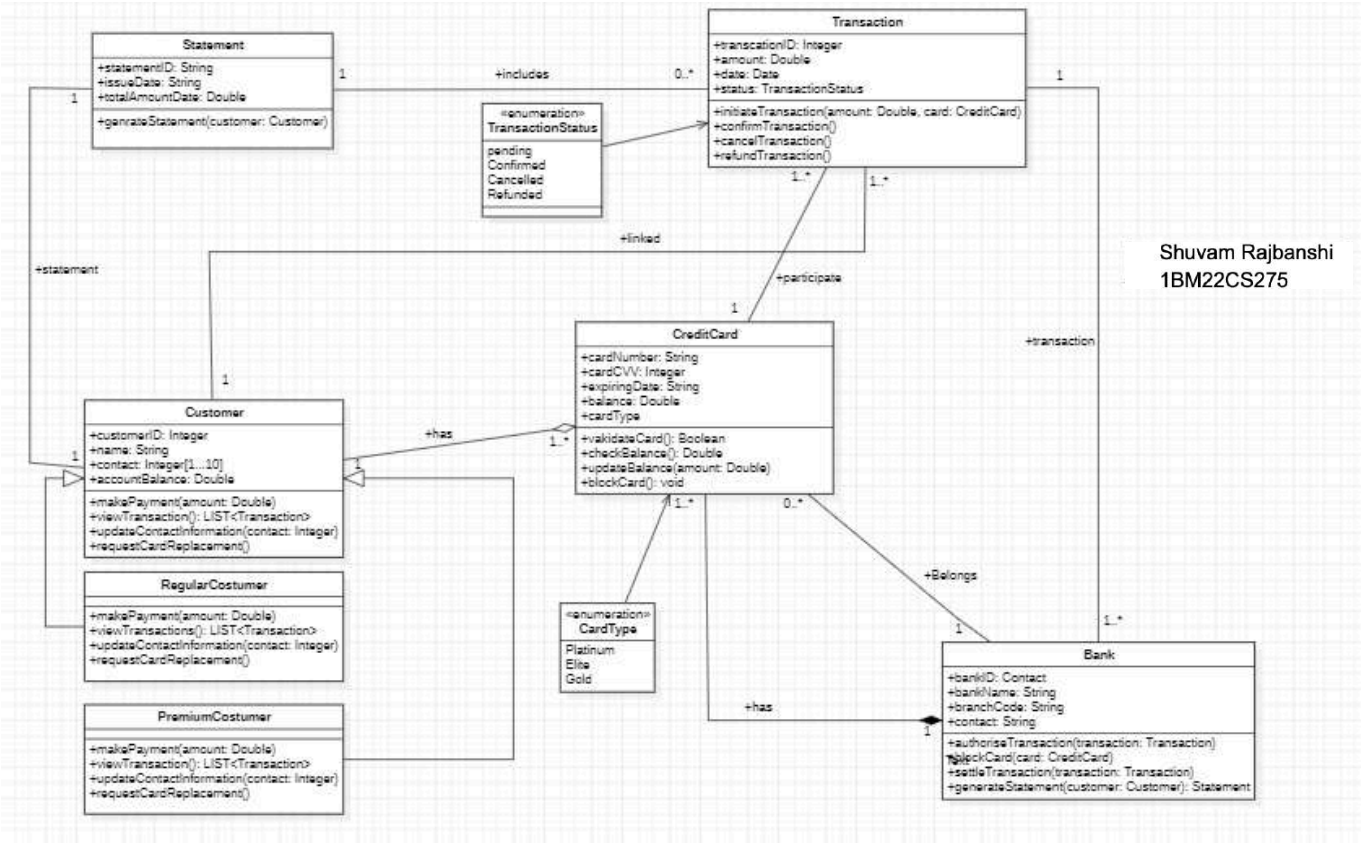
7. Non-Functional Attributes:

- Security: Strong encryption and secure data handling
- Scalability: Supports increasing transaction volumes.

8. Schedule and Budget

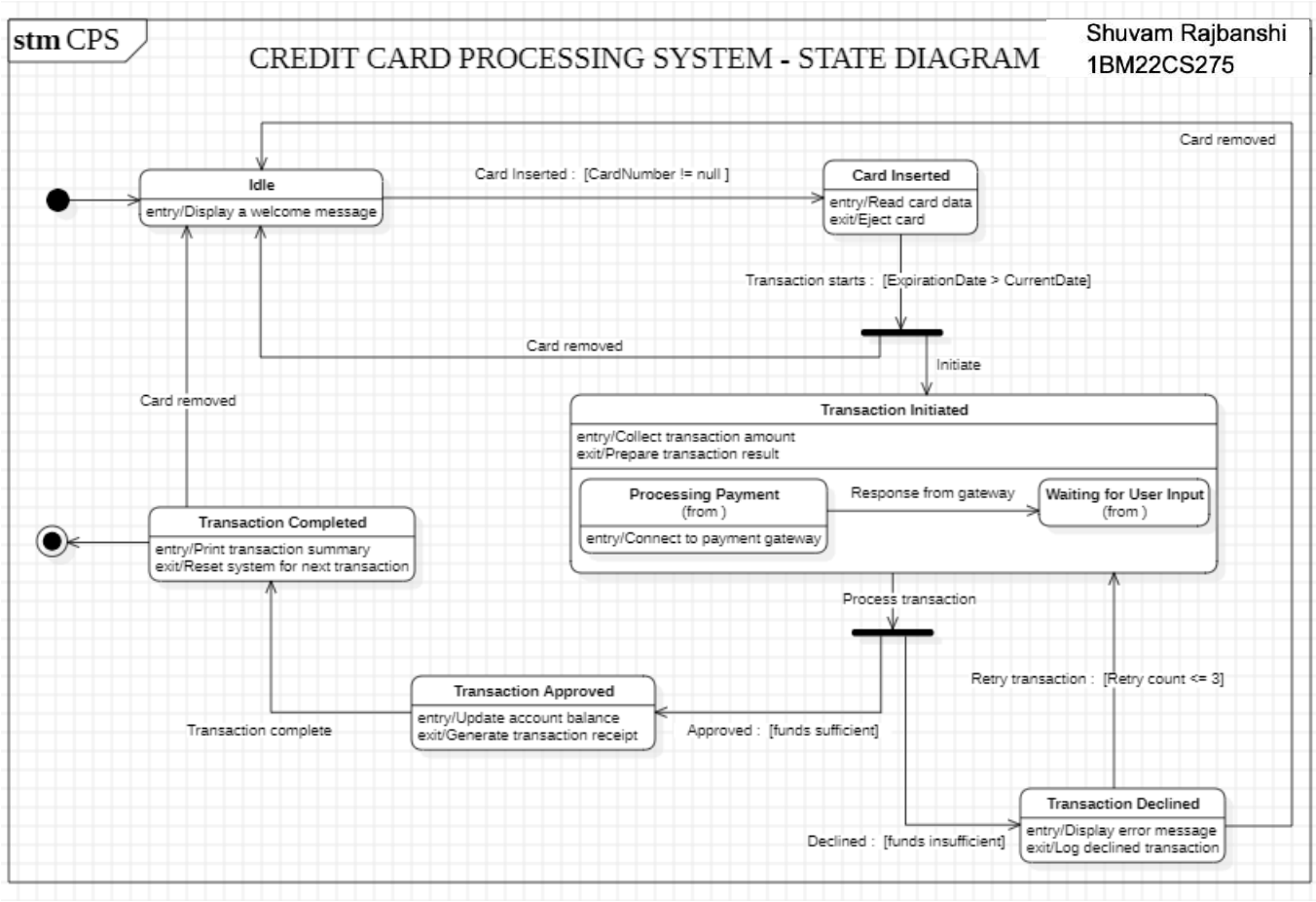
- Timeline: 6 months for development and testing
- Budget: Estimated $ 50,000

CLASS DIAGRAM



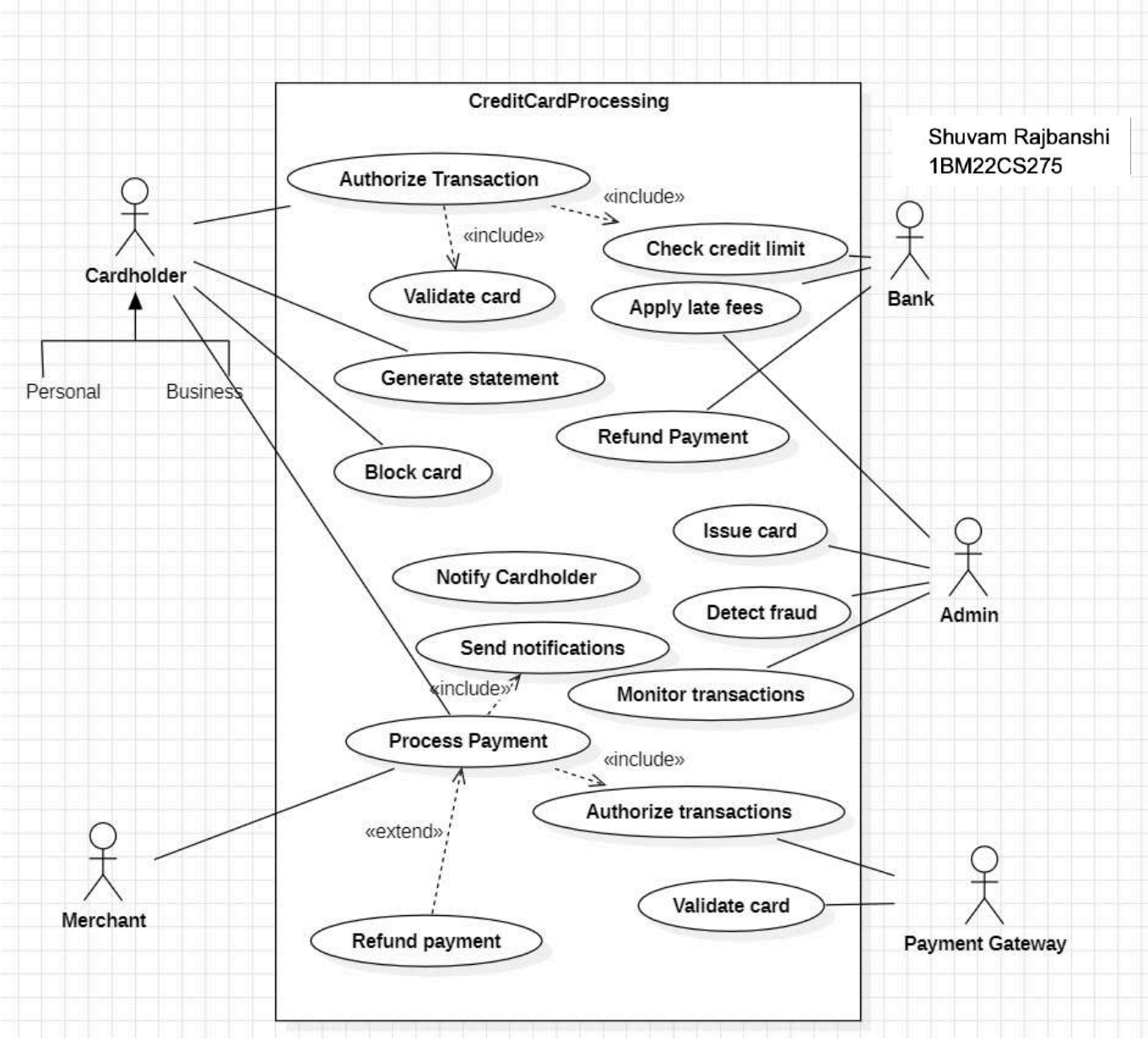
This UML class diagram illustrates the core components of a financial system. It depicts entities such as Customer, CreditCard, Transaction, Statement, and Bank, along with their attributes and relationships. Key features include transaction management, statement generation, and customer categorization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

STATE DIAGRAM



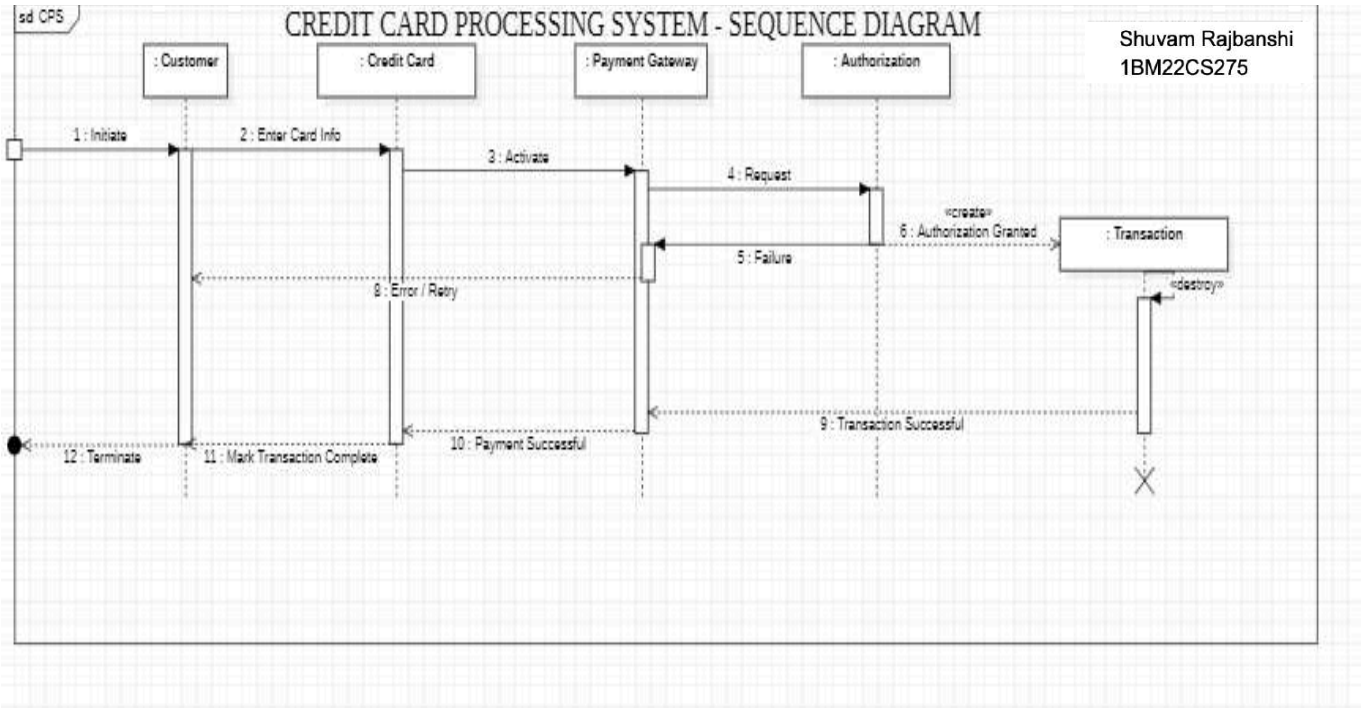
This state diagram models a credit card processing system. It shows the system transitioning through states like Idle, Transaction Initiated, Transaction Processing, and either Approved or Declined. The system processes card data, verifies transactions, and updates account balances accordingly.

USE CASE DIAGRAM



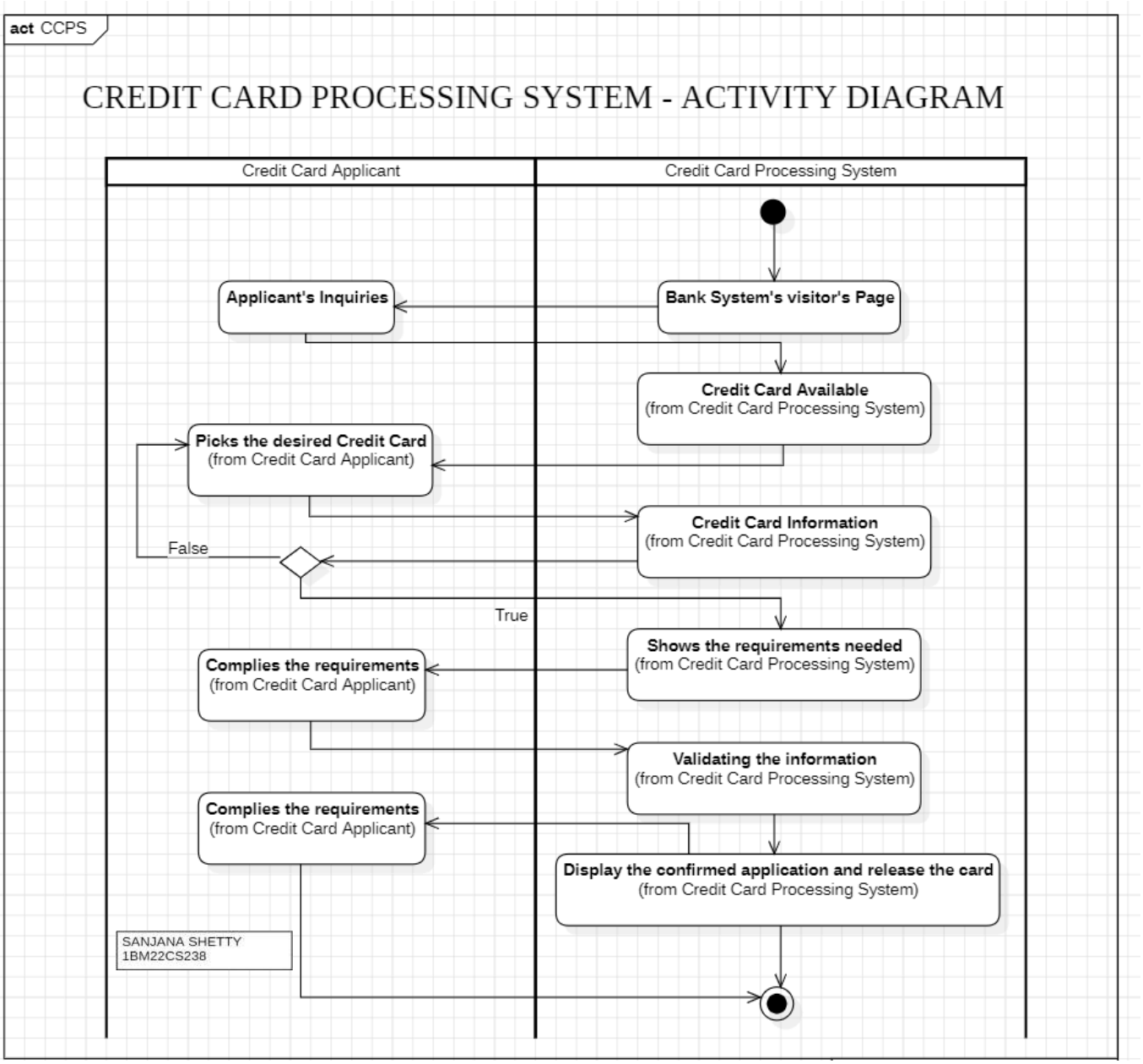
This UML use case diagram outlines the functionalities of a CreditCardProcessing system. It shows various actors like Cardholder (Personal and Business), Bank, Admin, Merchant, and Payment Gateway interacting with the system through different use cases. Key features include authorizing transactions, 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.

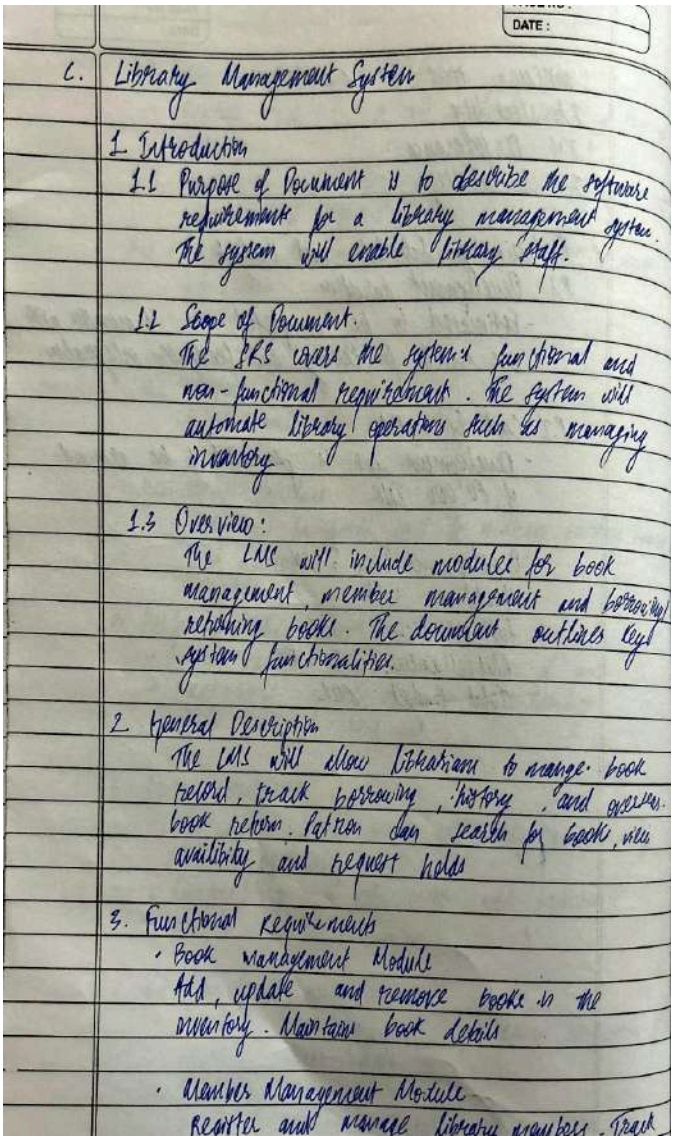
ACTIVITY DIAGRAM



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)



C. Library Management System

1. Introduction

1.1 Purpose of Document is to describe the software requirements for a library management system. The system will enable library staff.

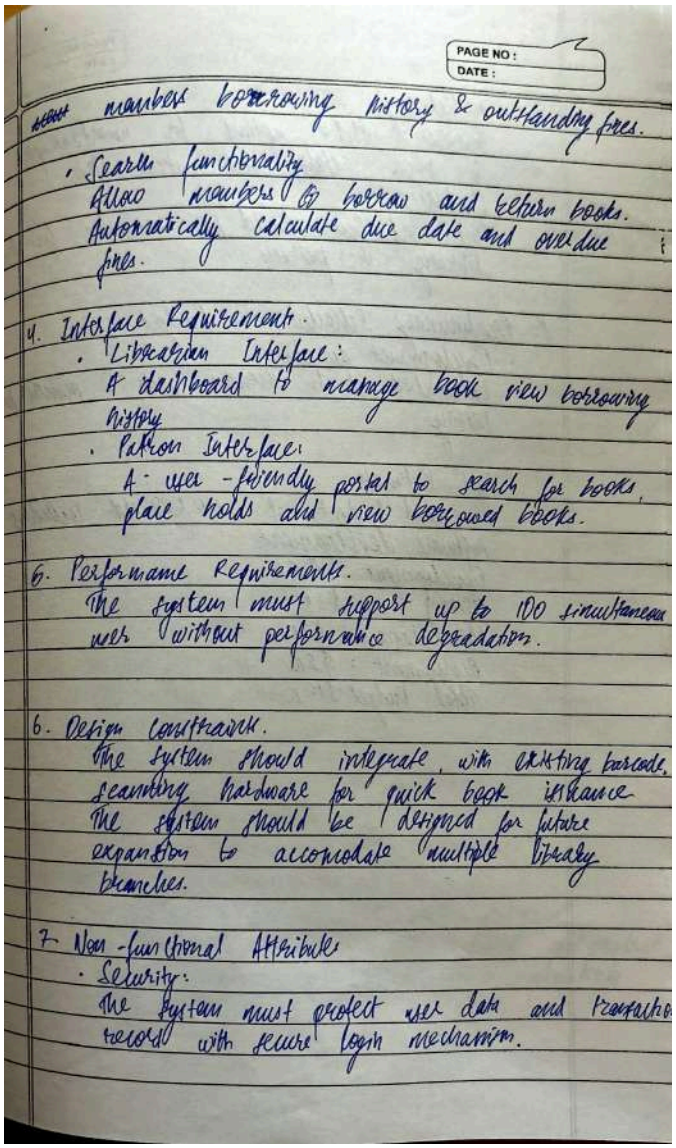
1.2 Scope of Document.
The SRS covers the system's functional and non-functional requirements. The system will automate library operations such as managing inventory

1.3 Overview:
The LMS will include modules for book management, member management and borrowing/returning books. The document outlines key system functionalities.

2. General Description
The LMS will allow librarians to manage book record, track borrowing history, and oversee book return. Patron can search for book, view availibity and request holds

3. Functional Requirements
• Book management Module
Add, update and remove books in the inventory. Maintain book details

• Member Management Module
Register and manage library members. Track



members borrowing history & outstanding fines.

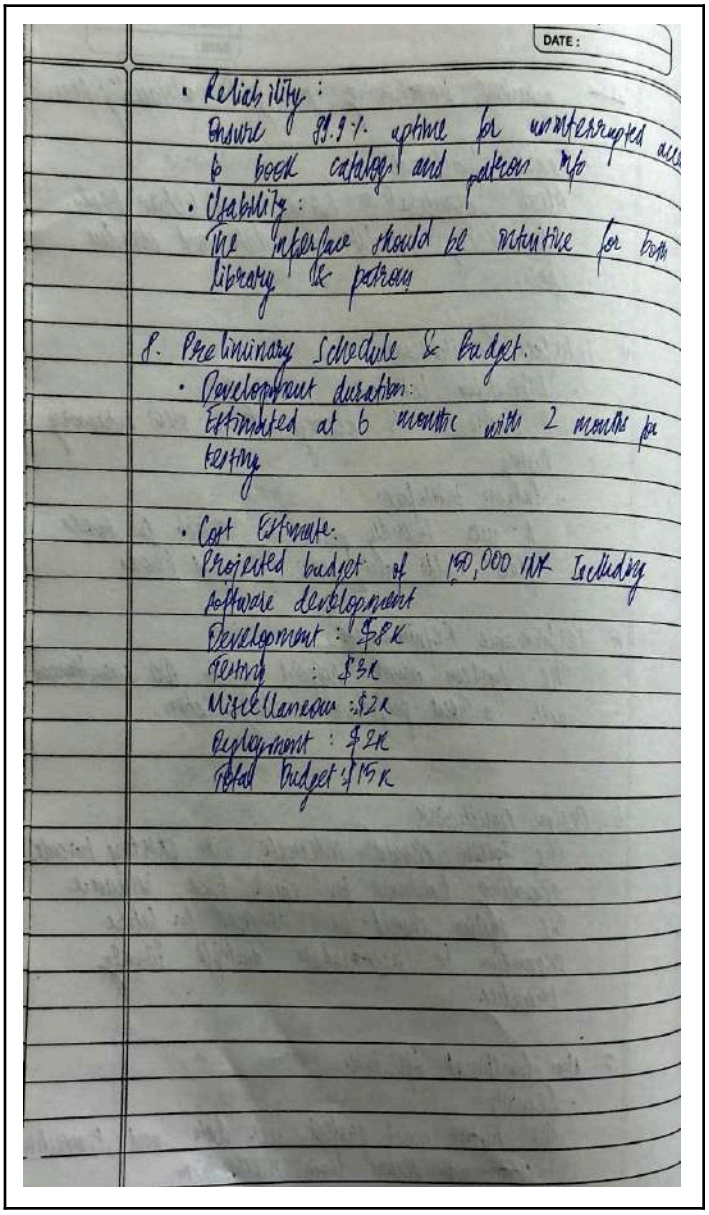
• Search functionality
Allow members to borrow and return books.
Automatically calculate due date and overdue fines.

4. Interface Requirement.
• Librarian Interface:
A dashboard to manage book view borrowing history
• Patron Interface:
A user-friendly portal to search for books, place holds and view borrowed books.

6. Performance Requirements.
The system must support up to 100 simultaneous users without performance degradation.

6. Design Constraints.
The system should integrate with existing barcode scanning hardware for quick book issuance
The system should be designed for future expansion to accomodate multiple library branches.

7. Non-functional Attribute
• Security:
The system must protect user data and transaction record with secure login mechanism.

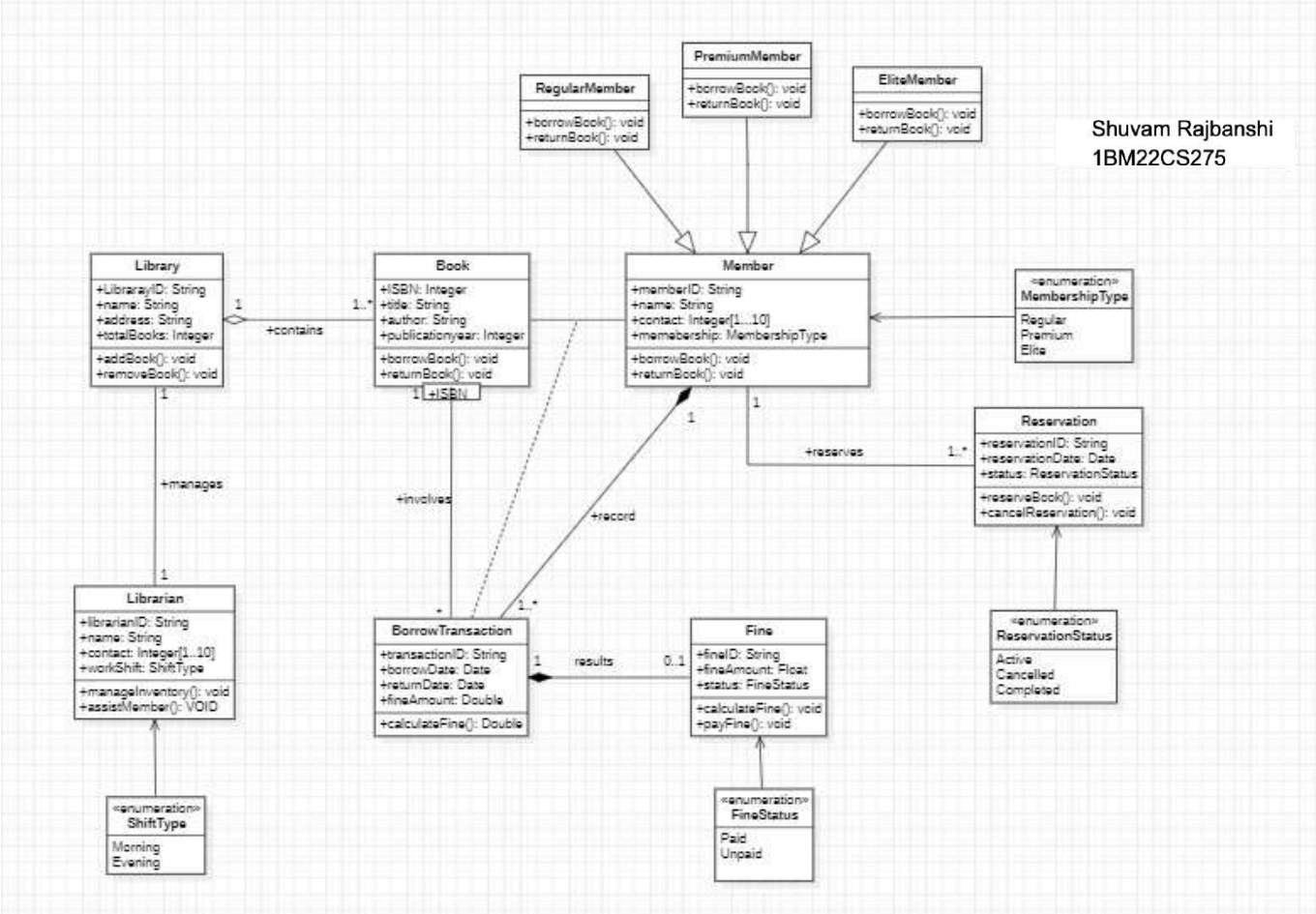


DATE :

- Reliability :
  Ensure 99.9% uptime for uninterrupted access to book catalogs and patron info
- Usability :
  The interface should be intuitive for both library & patrons

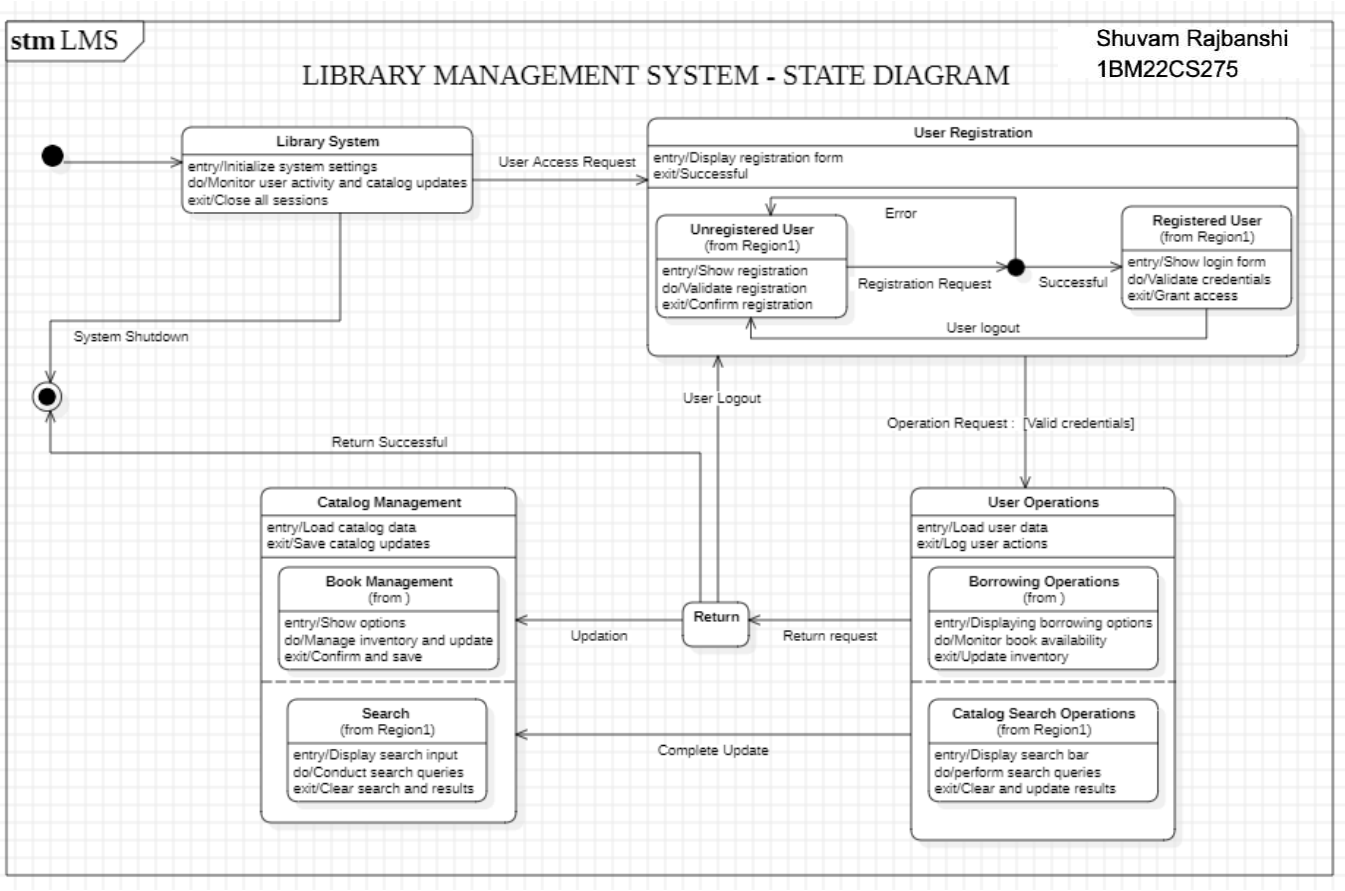
8. Preliminary Schedule & Budget.
- Development duration:
  Estimated at 6 months with 2 months for testing
- Cost Estimate:
  Projected budget of 150,000 BDT including software development
  Development : $8K
  Testing : $3K
  Miscellaneous : $2K
  Deployment : $2K
  Total budget : $15K

CLASS DIAGRAM



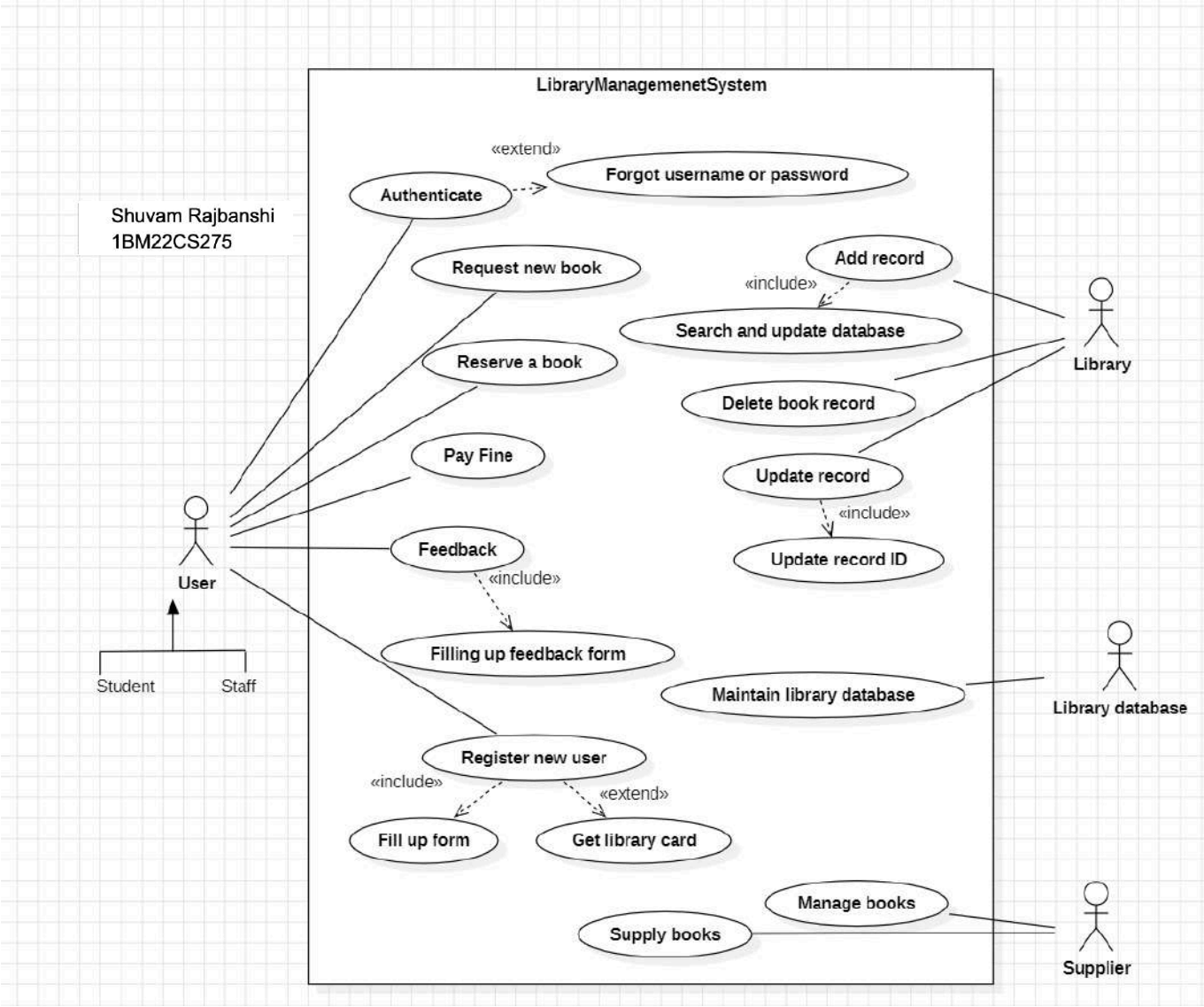
This UML class diagram illustrates the core components of a library management system. It depicts entities such as Library, Book, Member, Librarian, and their relationships. Key features include book reservations, borrowing transactions, and fine management. The diagram also includes enumerations for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

STATE DIAGRAM

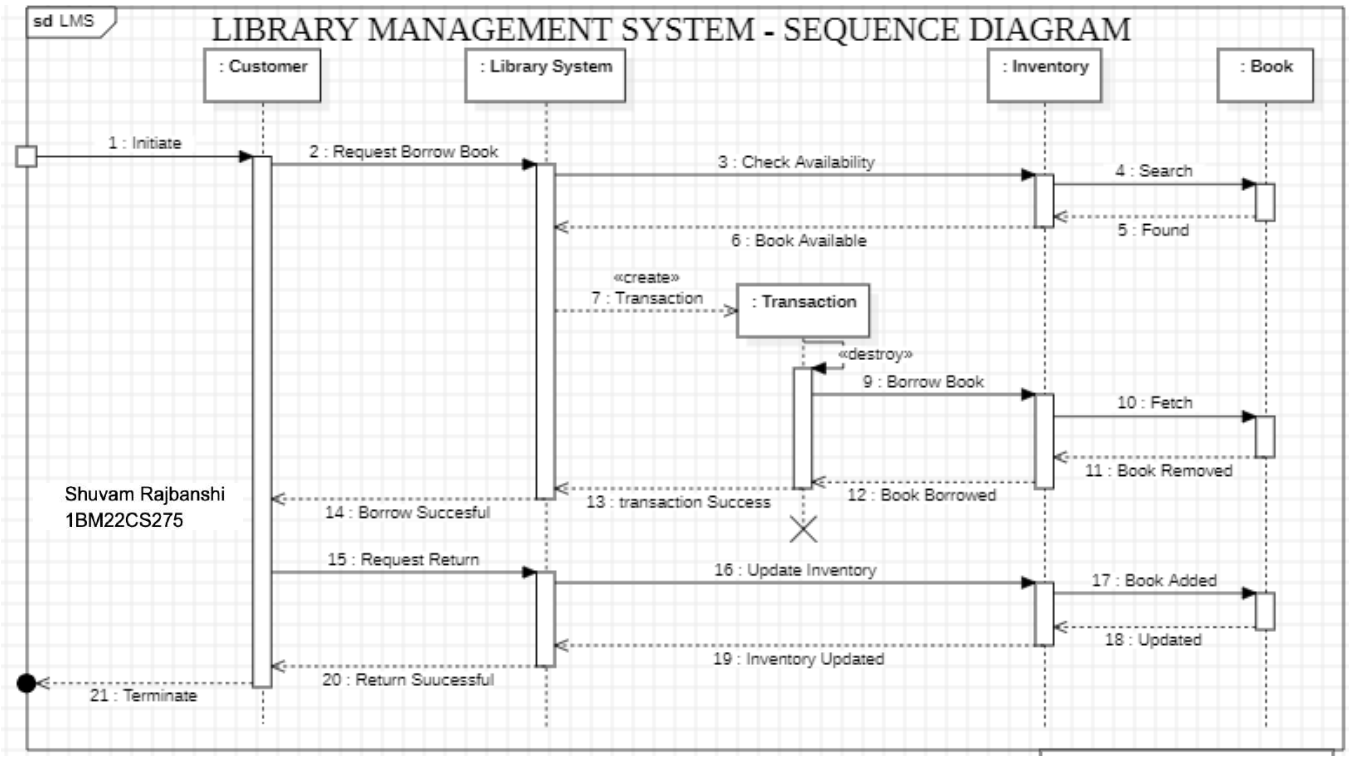


This state diagram illustrates the workflow of a library management system. It starts with the system initializing settings and monitoring user activity. Users can register and log in to access operations like catalog management, book management, borrowing operations, and catalog search. The system handles 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.

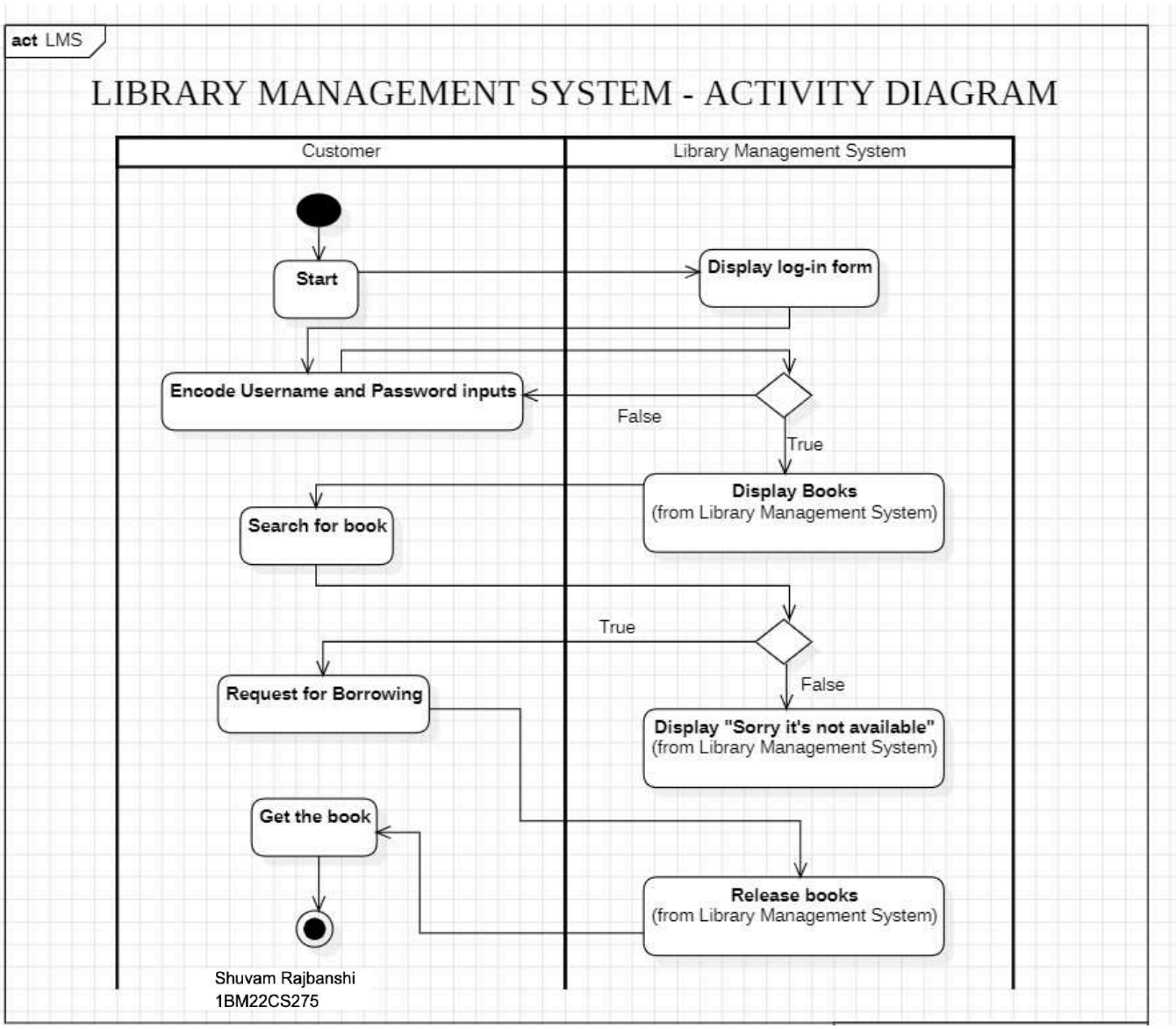
USE CASE DIAGRAM



SEQUENCE DIAGRAM

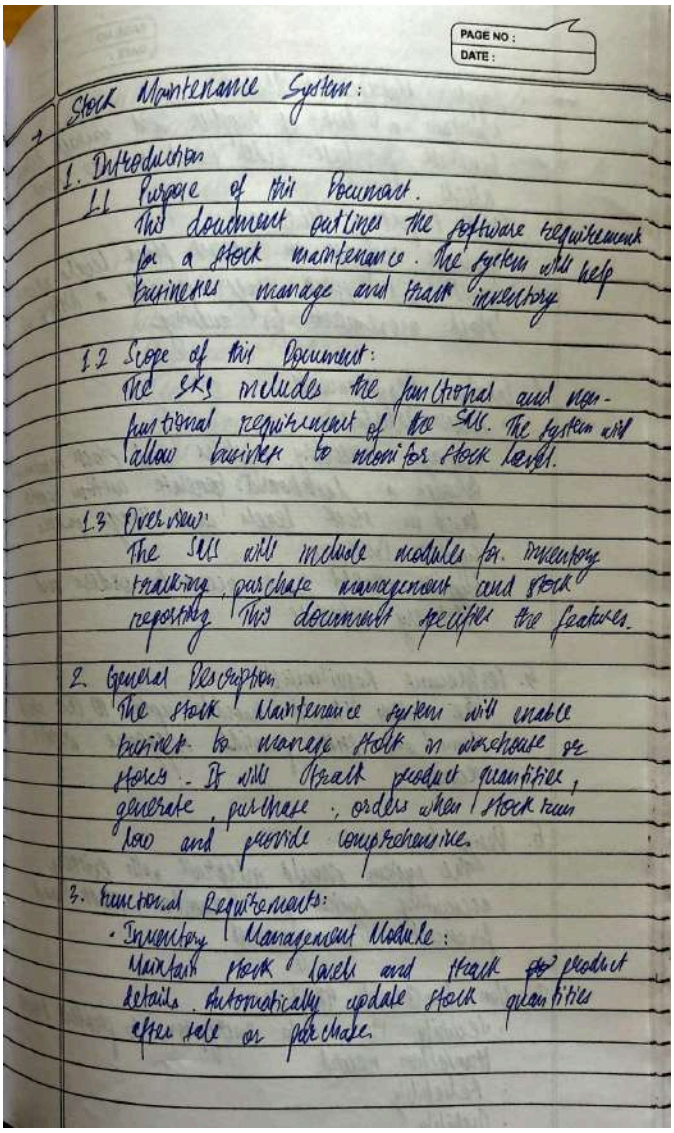


ACTIVITY DIAGRAM



STOCK MAINTENANCE SYSTEM

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)



Stock Maintenance System:

1. Introduction

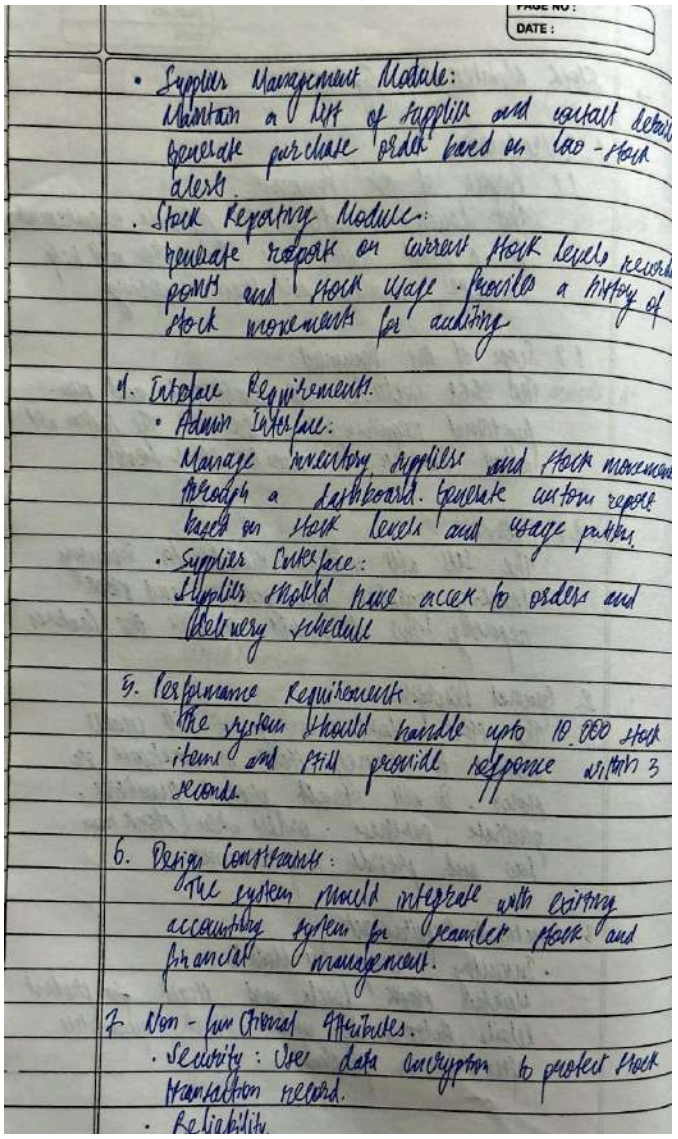
1.1 Purpose of this Document.
This document outlines the software requirement for a stock maintenance. The system will help businesses manage and track inventory

1.2 Scope of this Document:
The SRS includes the functional and non-functional requirement of the SMS. The system will allow business to monitor stock levels.

1.3 Overview
The SMS will include modules for inventory tracking, purchase management and stock reporting. This document specifies the features.

2. General Description
The stock Maintenance system will enable business to manage stock in warehouse or stores. It will track product quantities, generate purchase orders when stock run low and provide comprehensive.

3. Functional Requirements:
- Inventory Management Module:
Maintain stock levels and track for product details. Automatically update stock quantities after sale or purchase.



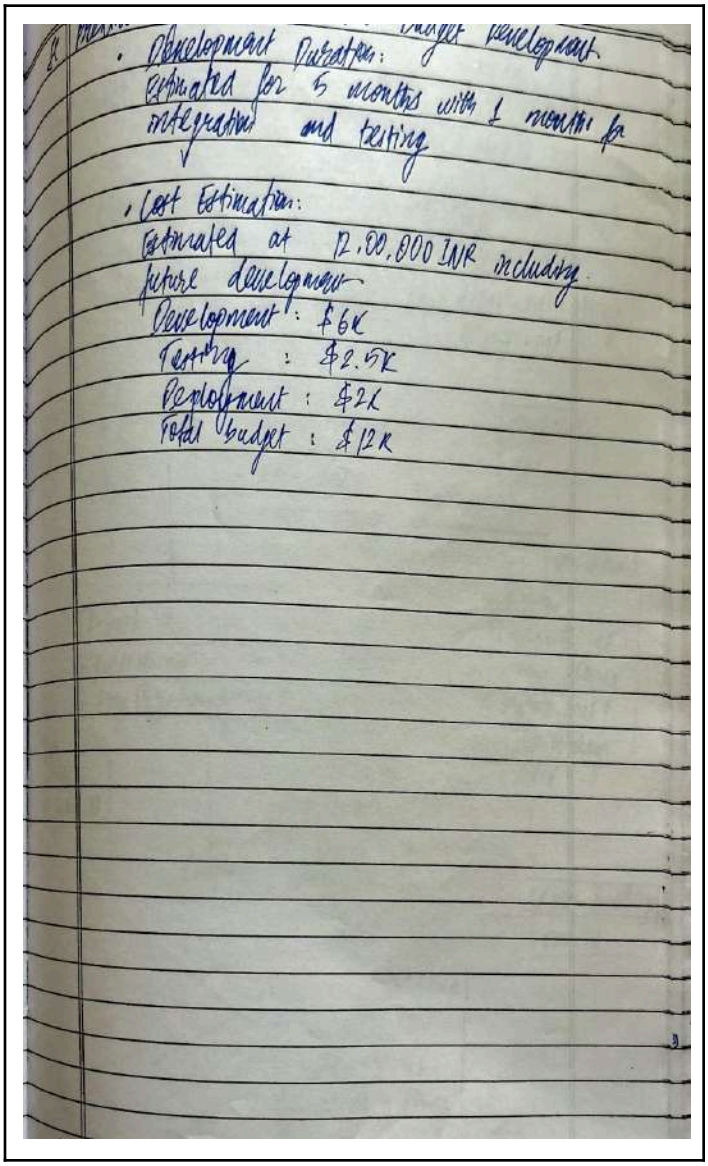
- Supplier Management Module:
Maintain a list of supplier and contact details generate purchase order based on low-stock alerts.
- Stock Reporting Module:
generate reports on current stock levels reorder points and stock usage. provides a history of stock movements for auditing

4. Interface Requirements.
- Admin Interface:
Manage inventory suppliers and stock movements through a dashboard. Generate custom reports based on stock levels and usage patterns.
- Supplier Interface:
Suppliers should have access to orders and delivery schedule

5. Performance Requirements.
The system should handle upto 10,000 stock items and still provide response within 3 seconds.

6. Design Constraints:
The system should integrate with existing accounting system for seamless stock and financial management.

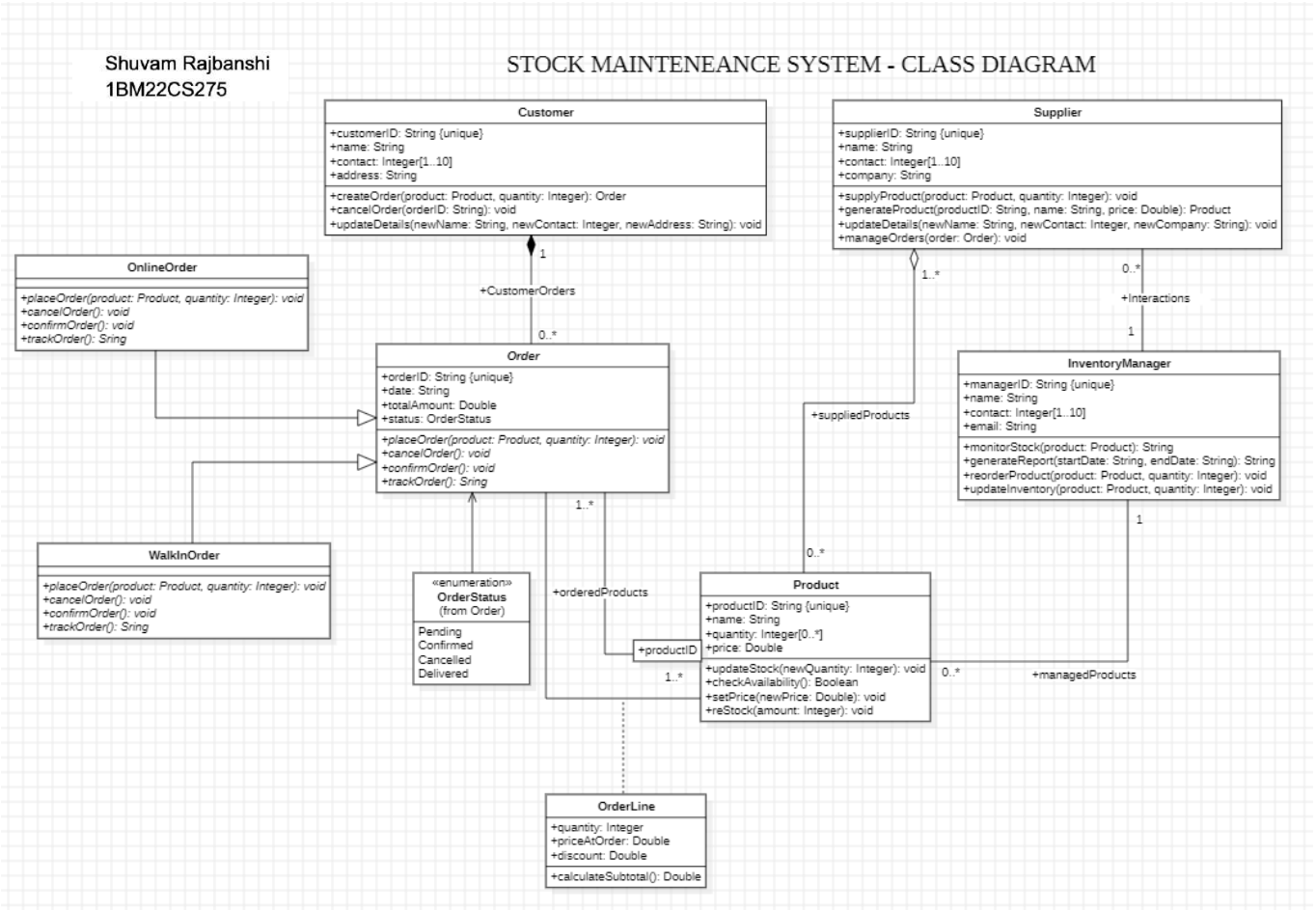
7. Non-functional Attributes.
- Security: Use data encryption to protect stock transaction record.
- Reliability



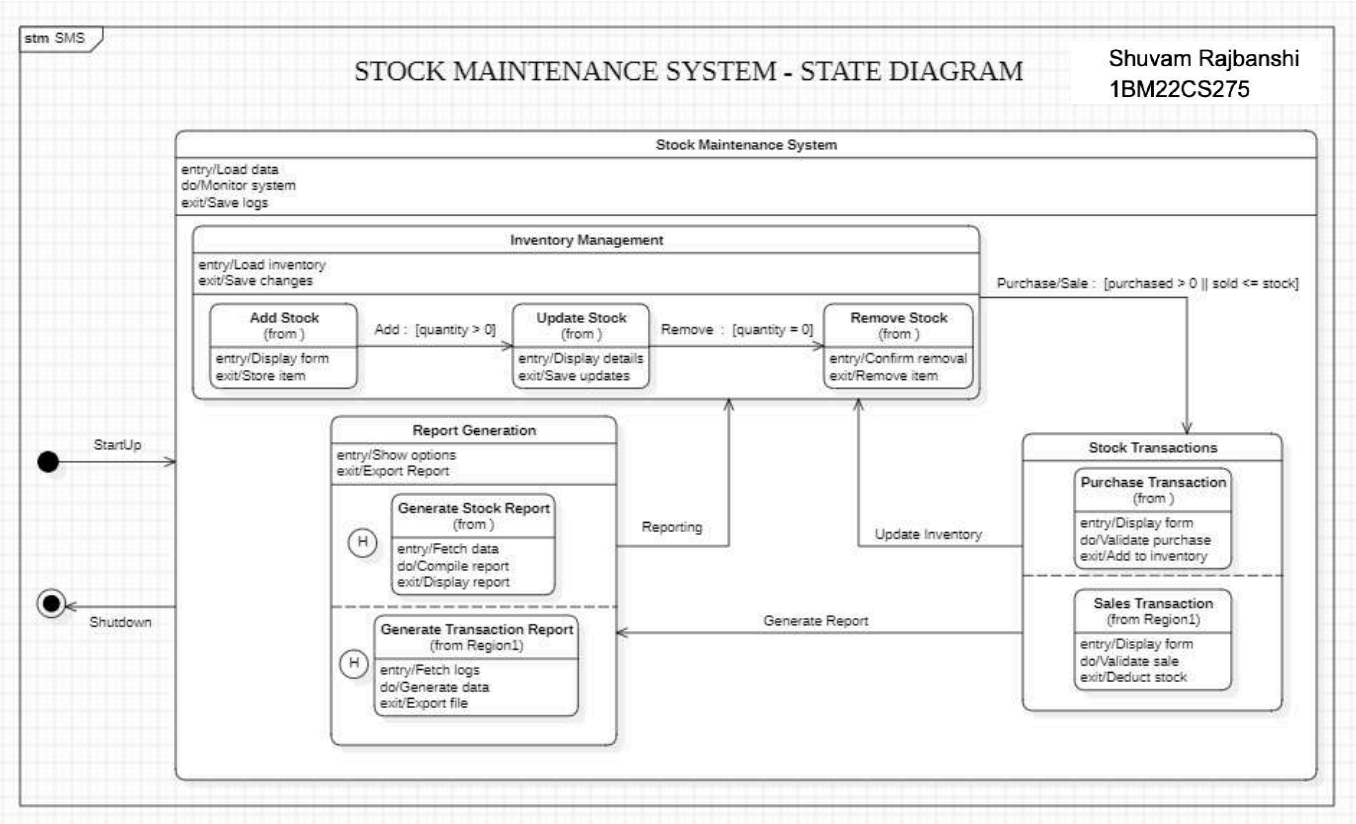
• Development Duration:
Estimated for 5 months with 1 month for integration and testing

• Cost Estimation:
Estimated at 12,00,000 INR including future development
Development : $6K
Testing : $2.5K
Deployment : $2K
Total Budget : $12K

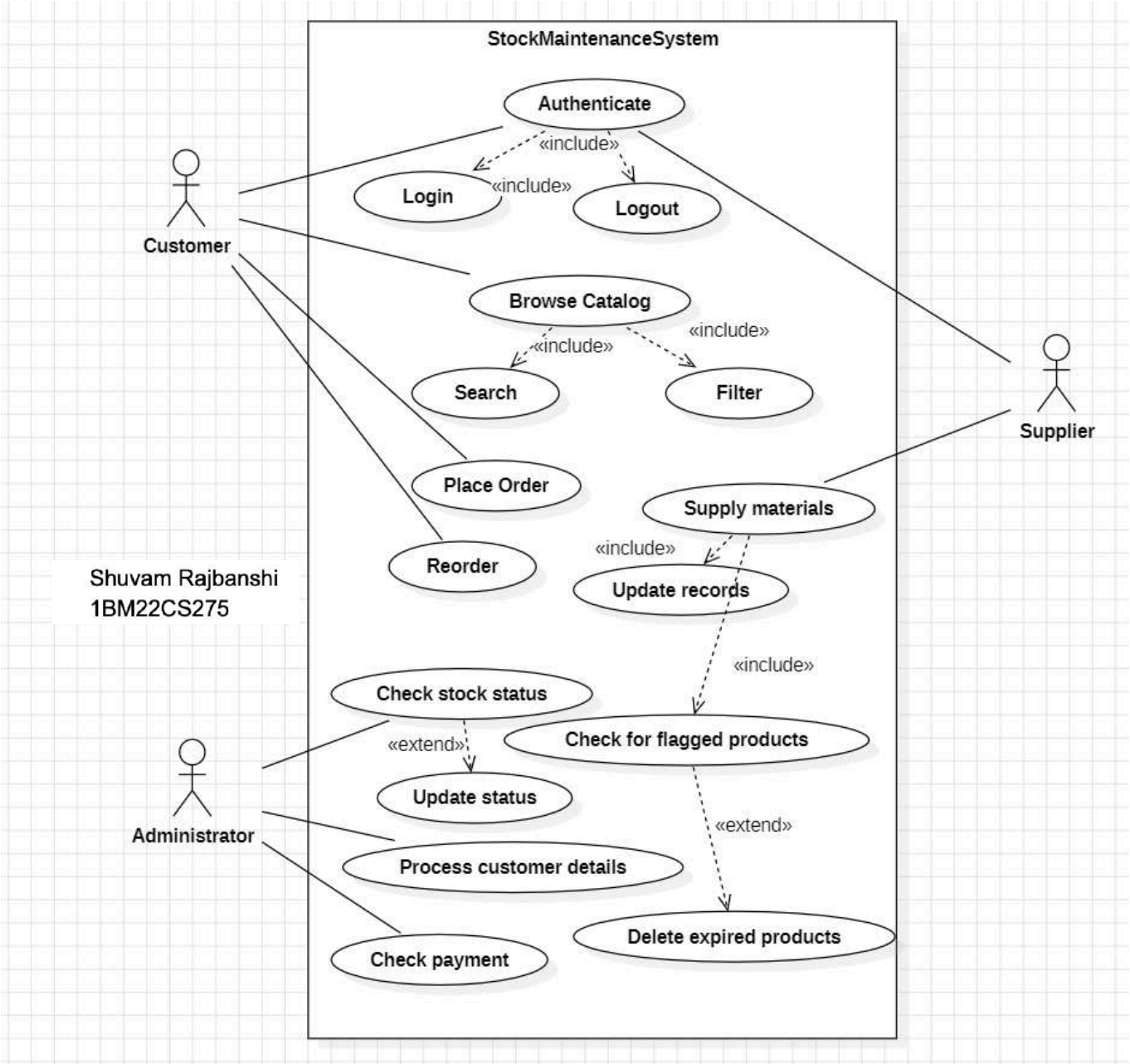
CLASS DIAGRAM



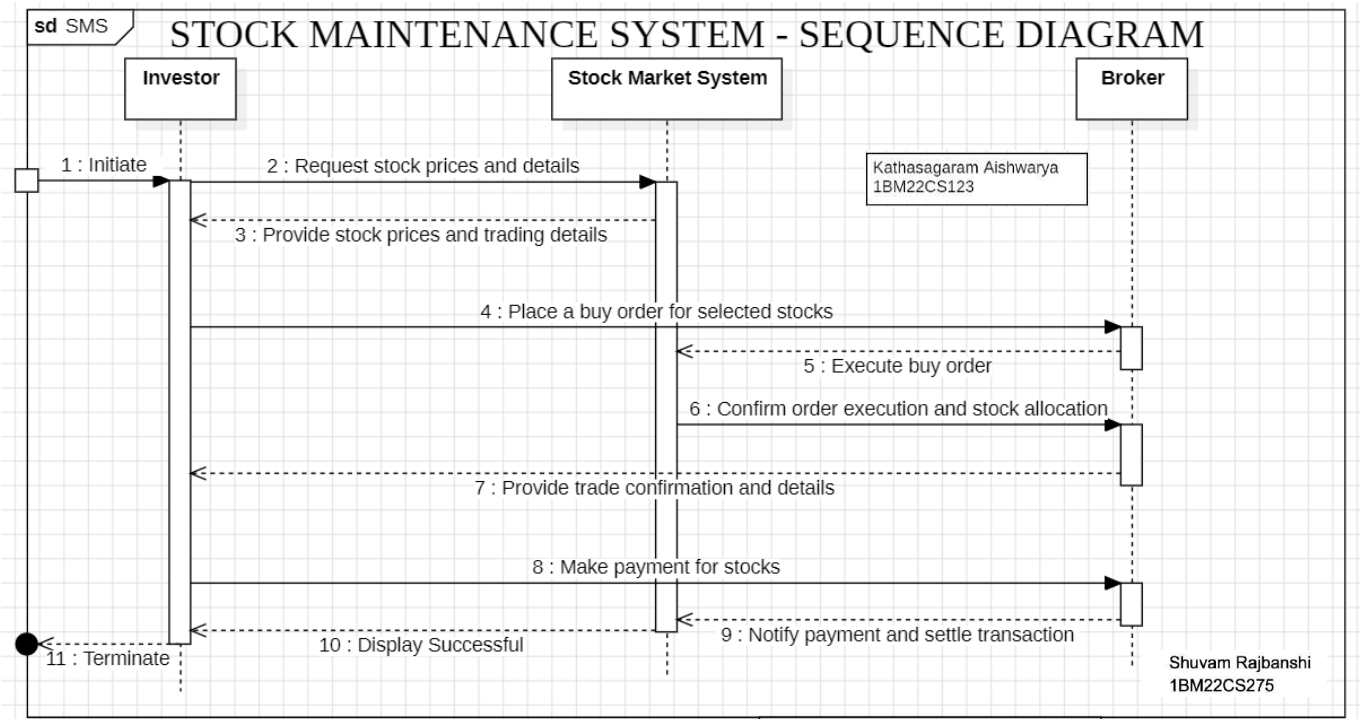
STATE DIAGRAM



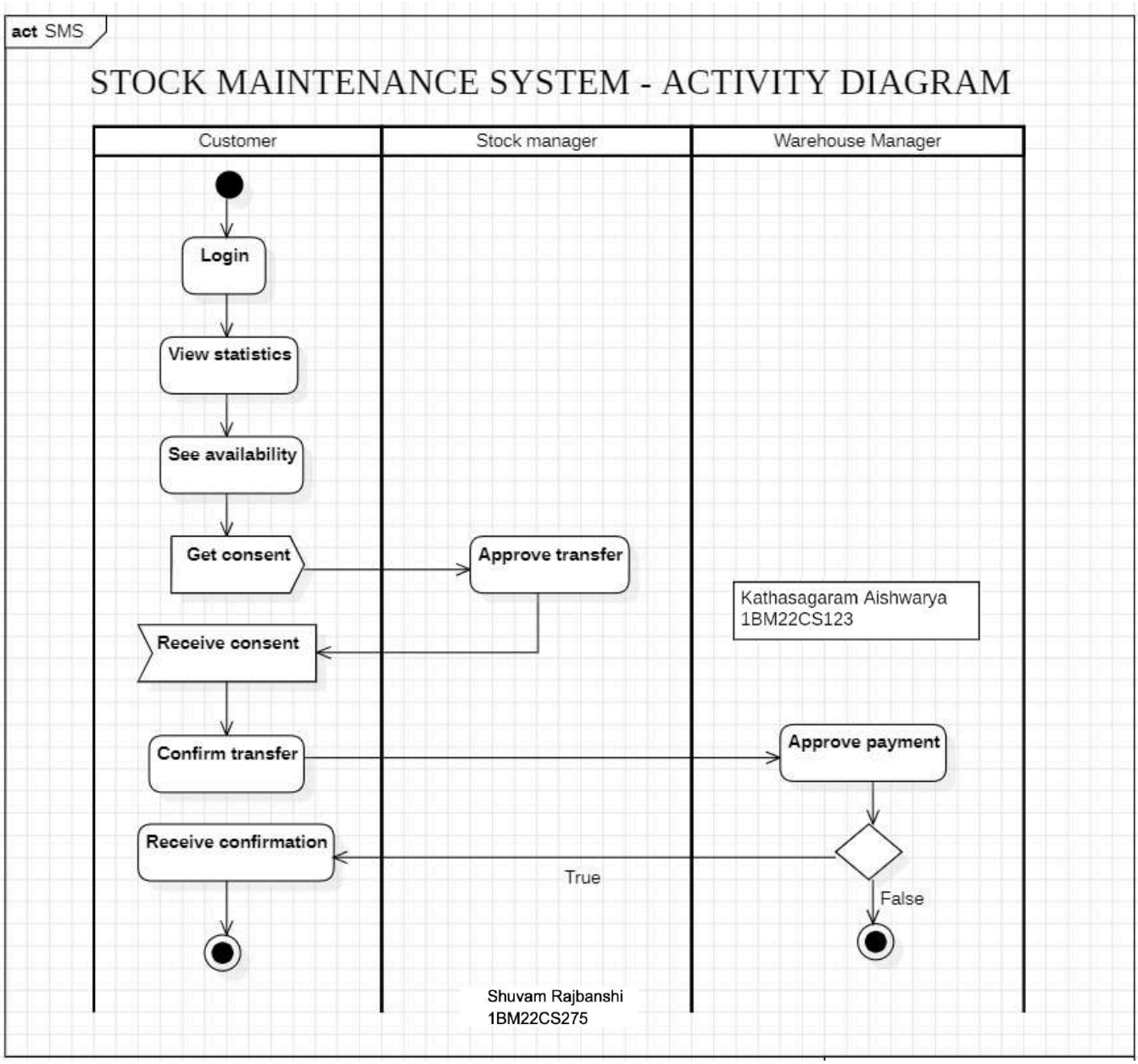
USECASE DIAGRAM



SEQUENCE DIAGRAM

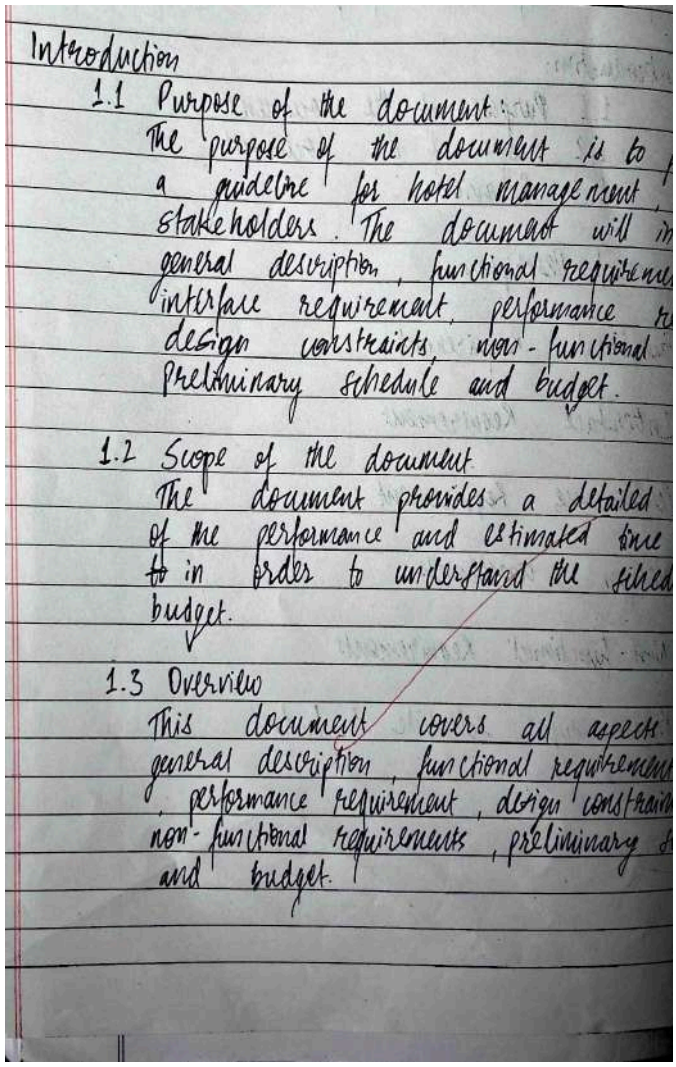


ACTIVITY DIAGRAM



PASSPORT AUTOMATION SYSTEM

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)



Introduction

1.1 Purpose of the document:

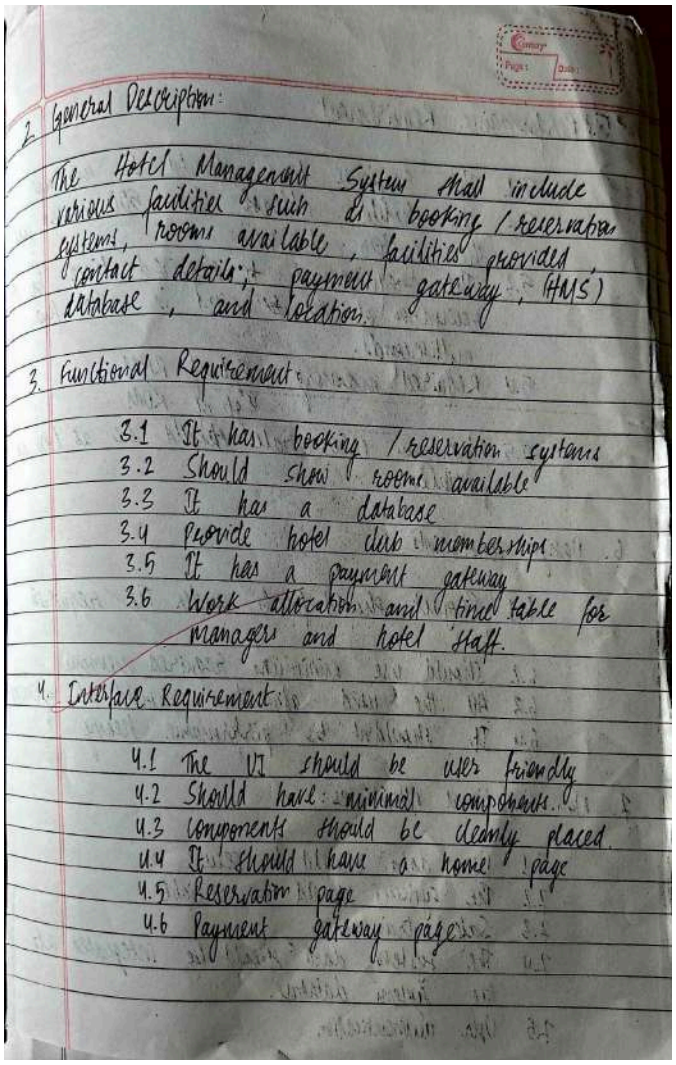
The purpose of the document is to
a guideline for hotel management
stakeholders. The document will
general description, functional requirem
interface requirement, performance r
design constraints, non-functional
Preliminary schedule and budget.

1.2 Scope of the document

The document provides a detailed
of the performance and estimated time
to in order to understand the fixed
budget.

1.3 Overview

This document covers all aspects
general description, functional requirem
, performance requirement, design constrain
non-functional requirements, preliminary
and budget.



2. General Description:

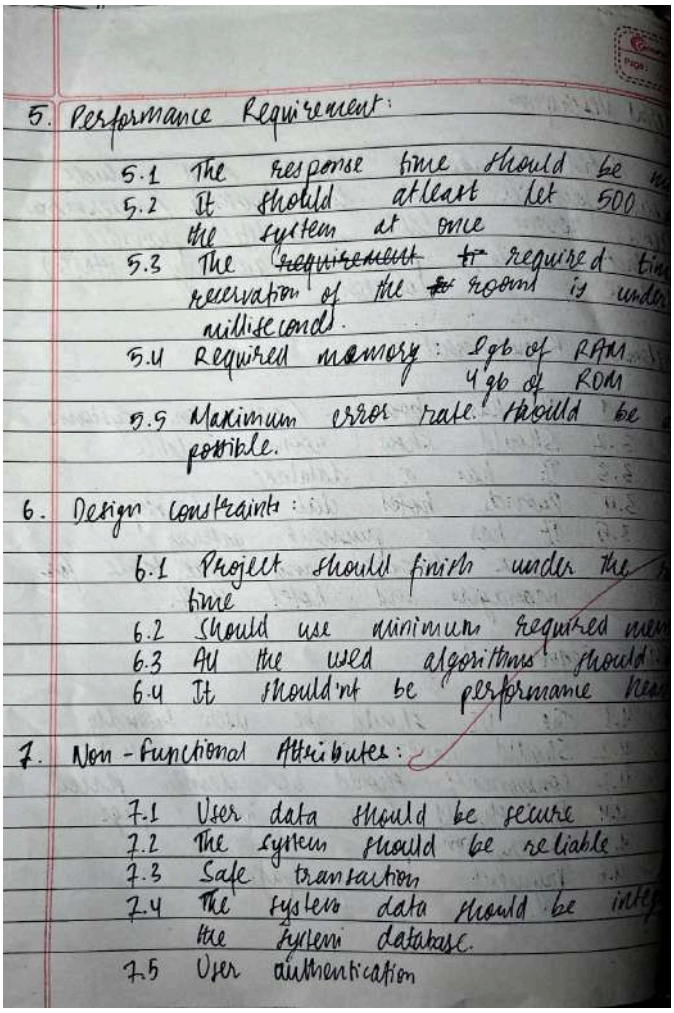
The Hotel Management System shall include
various facilities such as booking / reservation
systems, rooms available, facilities provided,
contact details, payment gateway, (HMS)
database, and location.

3. Functional Requirement:

3.1 It has booking / reservation systems
3.2 Should show rooms available
3.3 It has a database
3.4 Provide hotel club memberships
3.5 It has a payment gateway
3.6 Work allocation and time table for
managers and hotel staff.

4. Interface Requirement:

4.1 The UI should be user friendly
4.2 Should have minimal components.
4.3 Components should be clearly placed.
4.4 It should have a home page
4.5 Reservation page
4.6 Payment gateway page



5. Performance Requirement:

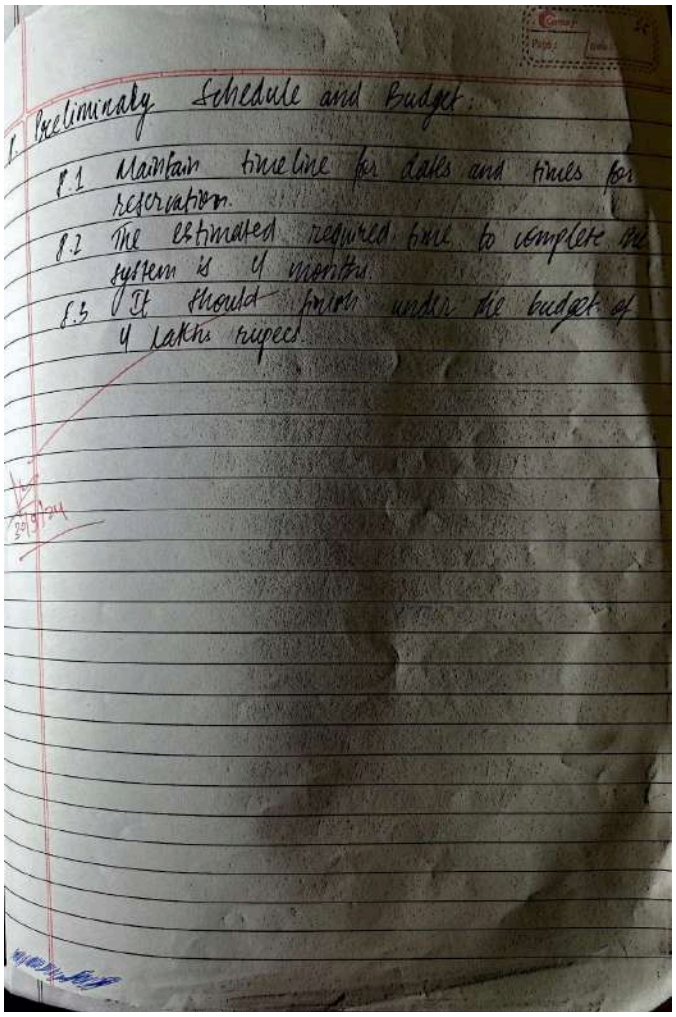
5.1 The response time should be
5.2 It should atleast let 500 the system at once
5.3 The ~~requirement~~ ~~it~~ required time reservation of the ~~the~~ rooms is under milliseconds.
5.4 Required memory: 2gb of RAM 4gb of ROM
5.5 Maximum error rate should be possible.

6. Design constraints:

6.1 Project should finish under the time
6.2 Should use minimum required mem
6.3 All the used algorithms should
6.4 It should'nt be performance hea

7. Non-functional Attributes:

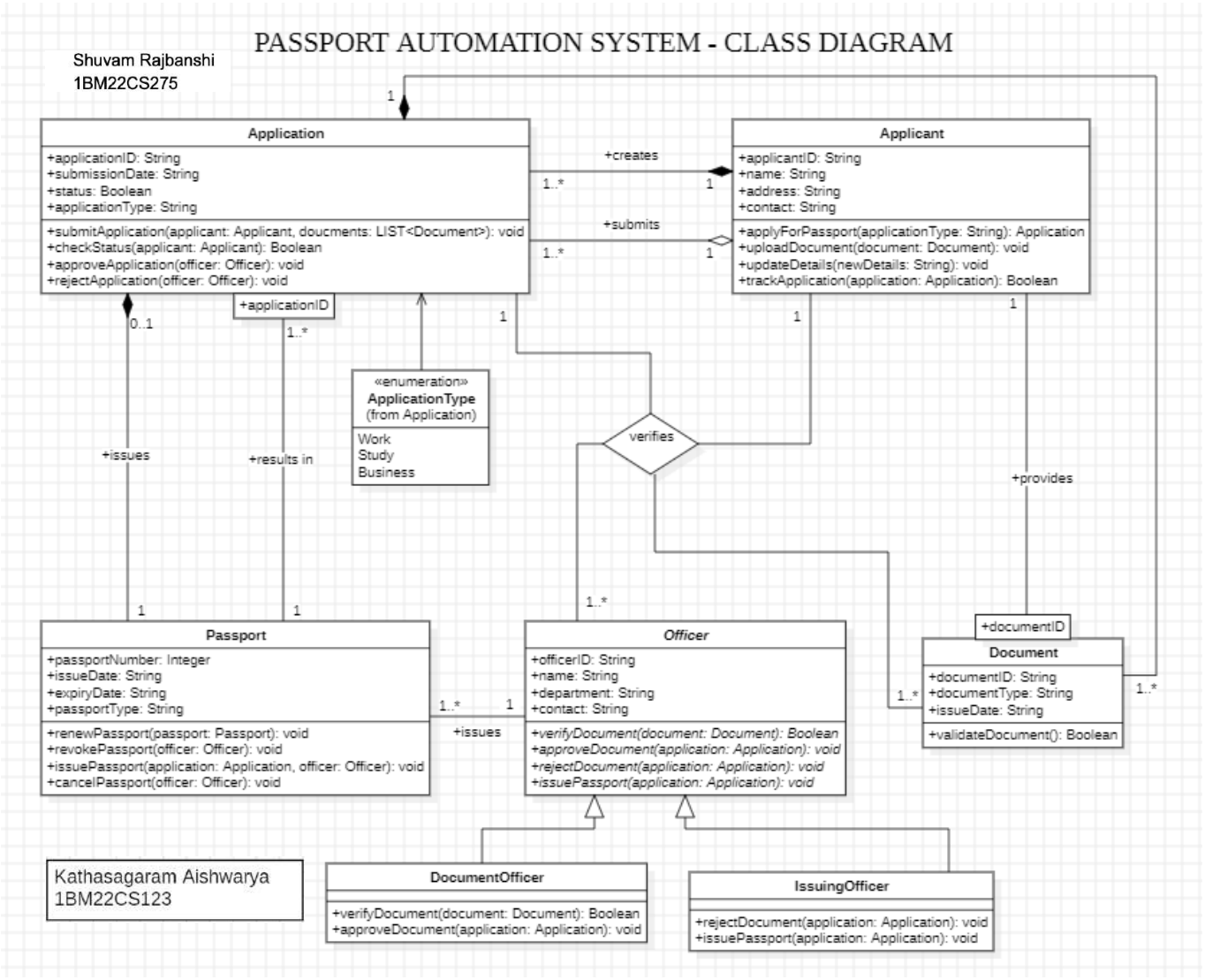
7.1 User data should be secure
7.2 The system should be reliable
7.3 Safe transaction
7.4 The system data should be integ the system database.
7.5 User authentication



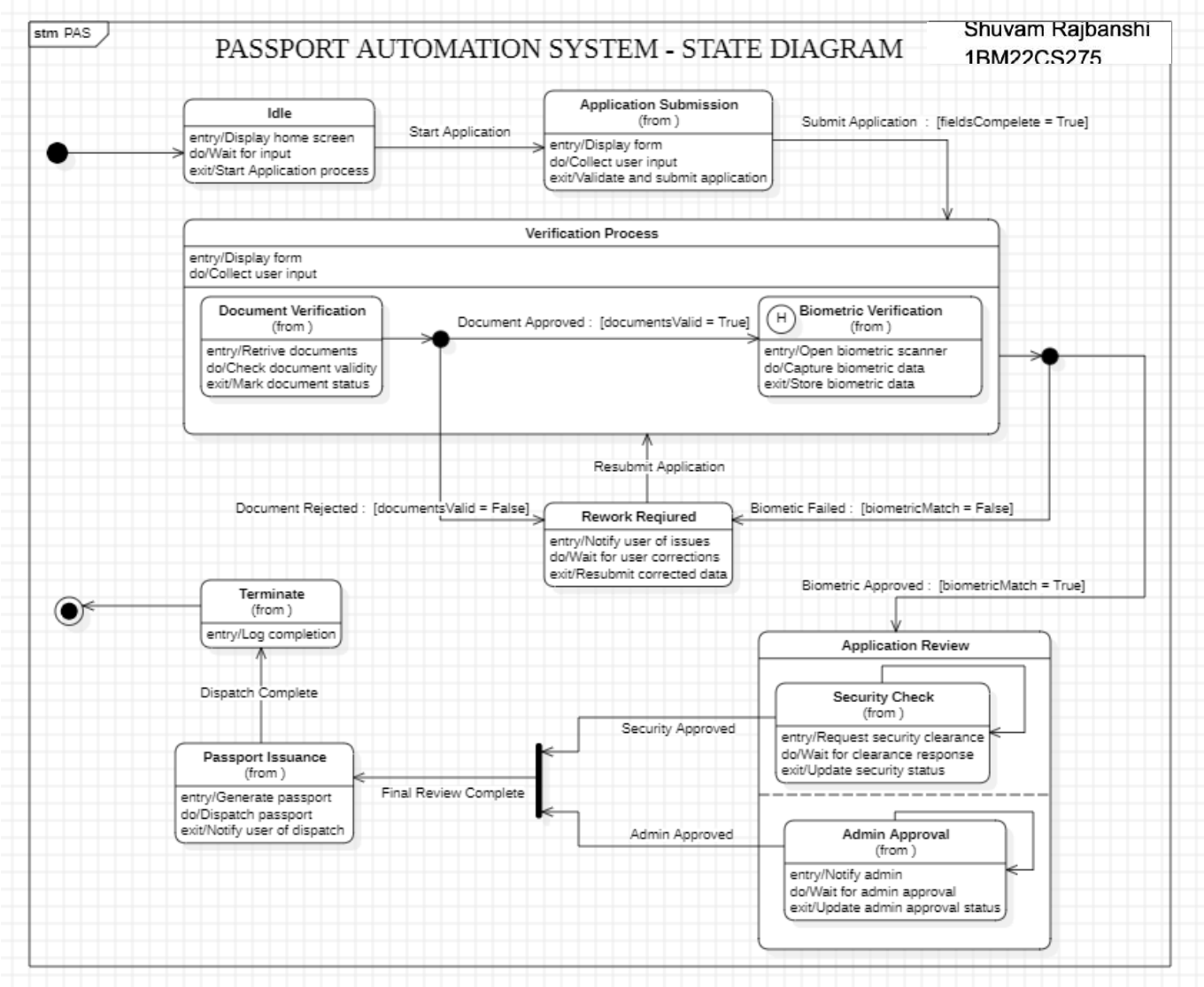
1. Preliminary Schedule and Budget:

8.1 Maintain timeline for dates and times for reservation.
8.2 The estimated required time to complete the system is 4 months.
8.3 It should finish under the budget of 4 Lakhs rupees.

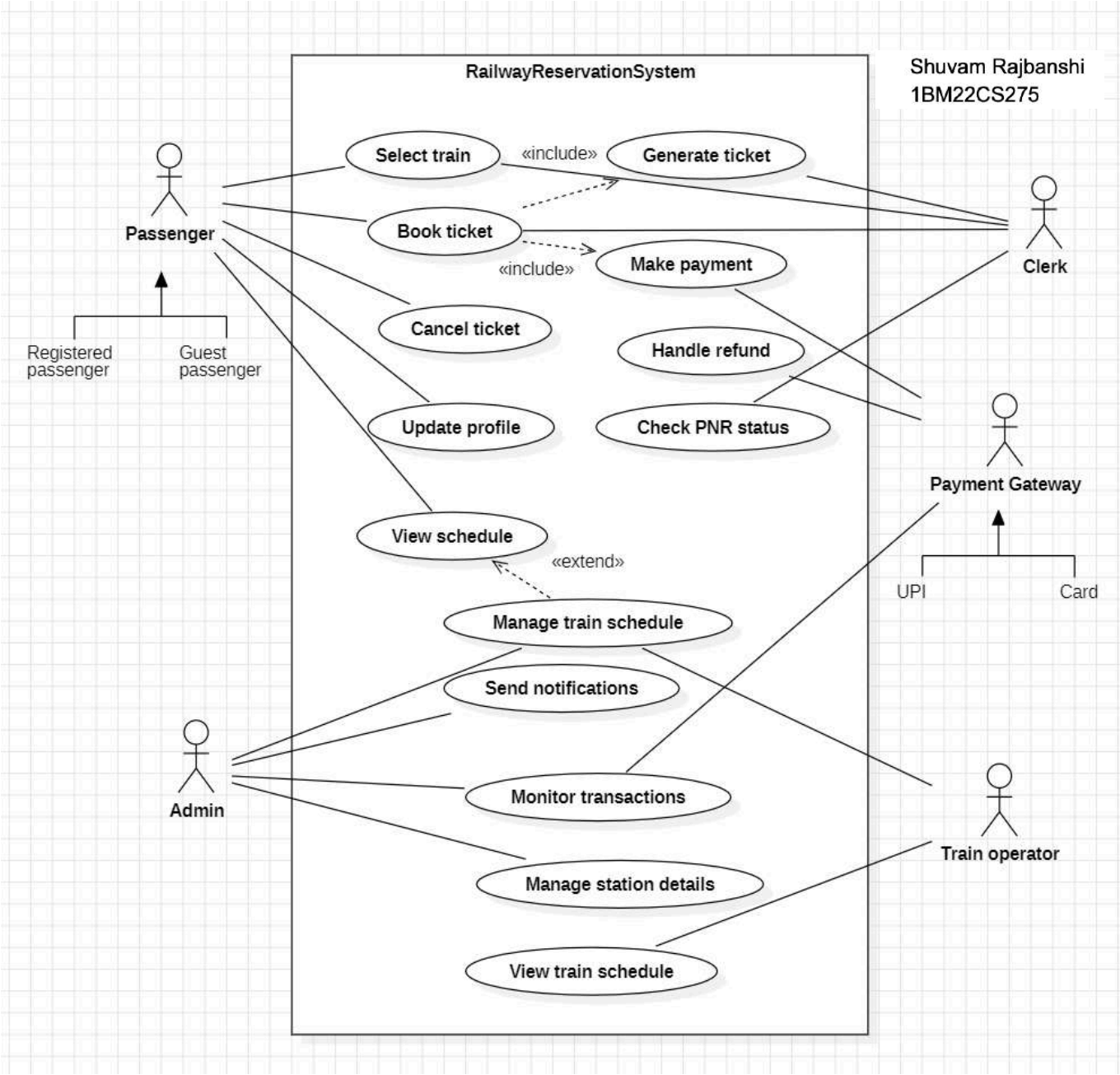
CLASS DIAGRAM



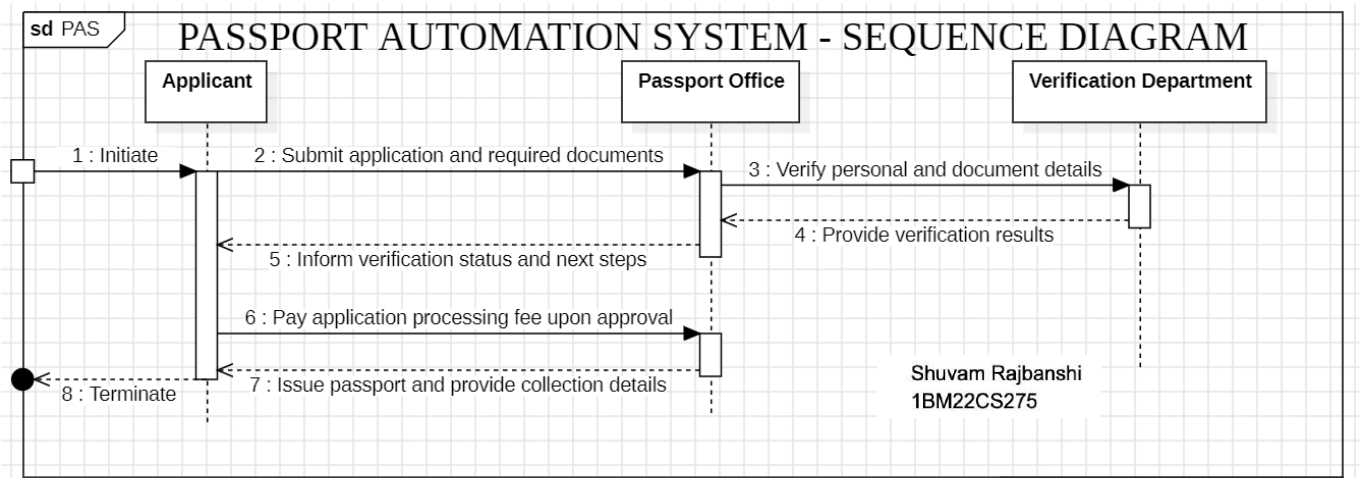
STATE DIAGRAM



USE CASE DIAGRAM



SEQUENCE DIAGRAM



ACTIVITY DIAGRAM