

0B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



Lab Record

Object Oriented Analysis and Design

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Technology
in
Computer Science and Engineering

Submitted by:

K L SRUJAN

(1BM24CS408)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Aug-Dec 2025

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



CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(23CS6PCSEO) laboratory has been carried out by **K L SRUJAN (1BM24CS408)** during the 5th Semester Aug-Dec-2025.

Signature of the Faculty Incharge:

NAME OF THE FACULTY: RoopaShree CS

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

Table of Contents

Sl no	content	Pg no
1.	Hotel Management System	4-9
2.	Credit Card Processing System	10-14
3.	Library Management System	15-18
4.	Stock Maintenance System	19-22
5.	Passport Automation System	23-26

1. Hotel Management System

1.1 SRS DOCUMENT:

LAB - 1	Date 19/8/25 Page _____
SRS Document is considered below in Application of Hotel Management System	
Problem Statement: Managing hotel manually or with outdated methods often lead to inefficiency, booking error, and poor customer service. A automated Hotel Management System with help to automate room booking, check-in/check-out, billing, staff management and Employee attendance.	
1.1 Introduction:	
1.1.1 purpose:	The purpose of this Document is to specify the software requirement of Hotel Management System. It is intended to provide a detailed description of functional and non-functional specification of the system for developers, tester and stakeholders.
1.1.2 Scope:	The HMS will automate hotel operations including room booking, customer check in/check out, billing,
1.2 Document Convention:	This document follows standard SRS conventions. Functional requirement are labeled as "FR" and non-functional requirements as "NFR". The document also uses UML diagrams and flowcharts for illustration.

Date _____ Page _____	Date _____ Page _____								
<p>Intended Audience & Reading Suggestion: This document is intended for hotel managers, developers, project stakeholders, and testers. It's recommended to begin with the "Overall Description" to understand the system's context before reviewing specific requirements.</p> <p>Project Scope: The system is designed to manage hotel operations, including room booking, guest check-ins/outs, room assignments, invoicing, and integrating payment methods. It will also have reporting capabilities to help the hotel staff manage booking and optimize room availability.</p> <p>References:</p> <ul style="list-style-type: none"> o Hotel Management Standards o ISO/IEC 9126 for software quality o PCI-DSS standards for payment integration 	<p>2 Overall Description:</p> <p>2.1 Product Perspective:</p> <ul style="list-style-type: none"> → The system is a standard standalone system but may integrate with third-party payment gateways and email/SMS APIs for notifications. It includes modules for: <ul style="list-style-type: none"> → Customer Registration & Booking → Room Allocation & Availability → Billing & Payment → Staff and Services Management → Reporting <p>2.2 Product Functions:</p> <ul style="list-style-type: none"> → Book rooms online/offline → Register/check-in/check-out guests → Manage room inventory → Generate invoices and accept payment → Assign and manage staff tasks → Generate reports on occupancy, revenue etc. <p>2.3 User classes and characteristics:</p> <table border="0"> <tr> <td>User Role</td> <td>Description</td> </tr> <tr> <td>Receptionist</td> <td>Handle bookings, check-ins/outs</td> </tr> <tr> <td>Customer</td> <td>View/book rooms, make payment</td> </tr> <tr> <td>Staff</td> <td>Manage assigned tasks</td> </tr> </table> <p>2.4 Operating Environment:</p> <ul style="list-style-type: none"> → Web server (Apache/Tomcat) → Application Database (MySQL) → Browser-based user (Chrome, Firefox etc.) 	User Role	Description	Receptionist	Handle bookings, check-ins/outs	Customer	View/book rooms, make payment	Staff	Manage assigned tasks
User Role	Description								
Receptionist	Handle bookings, check-ins/outs								
Customer	View/book rooms, make payment								
Staff	Manage assigned tasks								

Date _____ Page _____	Date _____ Page _____
<p>2.5 Design & Implement Constraints:</p> <ul style="list-style-type: none"> → The system must comply with local data protection regulation → Use of secure communication (HTTPS) → Use of responsive UI design <p>2.6 User Documentation:</p> <ul style="list-style-type: none"> → User manual (PDF, online help) → Admin guide → Installation & deployment guide <p>2.7 Assumptions and Dependencies:</p> <ul style="list-style-type: none"> → Reliable internet connection is assumed for online access → External APIs are available and functional <p>3 Specific Requirements:</p> <p>3.1 Functional Requirements:</p> <p>FR 1: User Registration and login</p> <ul style="list-style-type: none"> → The system shall allow users to register using email / phone number → The system shall authenticate the user. <p>FR 2: Room Booking</p> <ul style="list-style-type: none"> → System shall display available room based on entered date range → System shall allow user to book room. → System shall allow update room availability. 	<p>FR 3: Check-in / check-out process</p> <ul style="list-style-type: none"> → The system shall allow receptionists to perform check-ins and check-outs → The system shall update the room status accordingly <p>FR 4: Billing and Payments</p> <ul style="list-style-type: none"> → The system shall generate bills for each stay and allow task assignment to staff members <p>FR 5: Staff Management</p> <p>The system shall allow the admin to add/edit/delete staff and task assignment to staff members</p> <p>FR 6: Reports</p> <p>The system shall generate daily, weekly, and monthly reports on occupancy and revenue</p> <p>3 Specific Requirements:</p> <p>3.1 Functional Requirements:</p> <p>FR 1: User Registration and login</p> <p>FR 2: Room Booking</p> <p>3.2 External Interface Requirements:</p> <p>3.2.1 User Interfaces:</p> <ul style="list-style-type: none"> → Responsive <p>3.2.2 Hardware Interface</p> <p>3.2.3 Software Interface</p> <p>3.2.4 Communication Interface</p> <p>3.3 Performance Requirements:</p> <p>The system shall support up to 100 concurrent users and response time shall be less than 3 sec.</p>

~~non functional~~

3.4 Security Requirements

→ security walls and anti

3.5 Extended Interface requirements

→ System must integrate with third-party travel agencies

→ integration with external payment gateway

→ ...

4 Appendices

→ UML diagram for system Architecture & use case

→ flow chart for the guest check in / check out process

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

→ ...

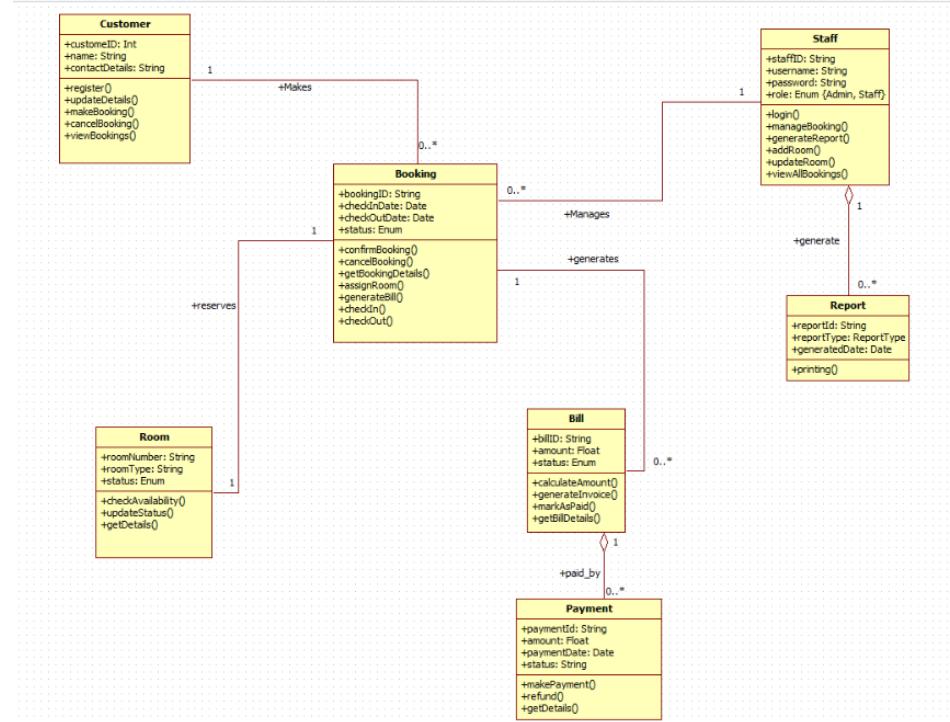
→ ...

→ ...

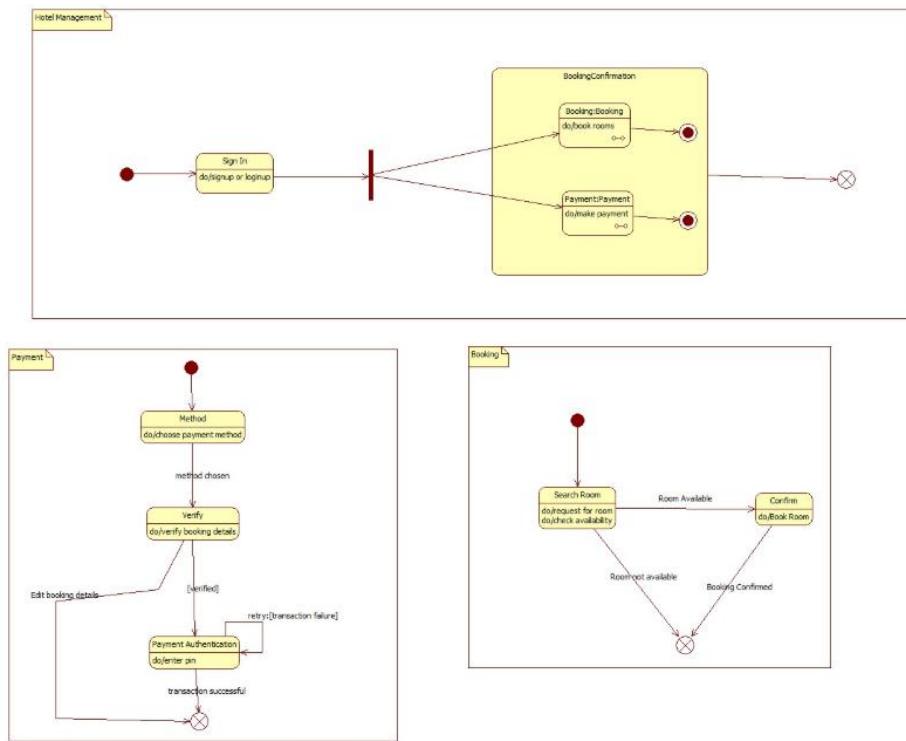
→ ...

→ ...

1.2 CLASS DIAGRAM

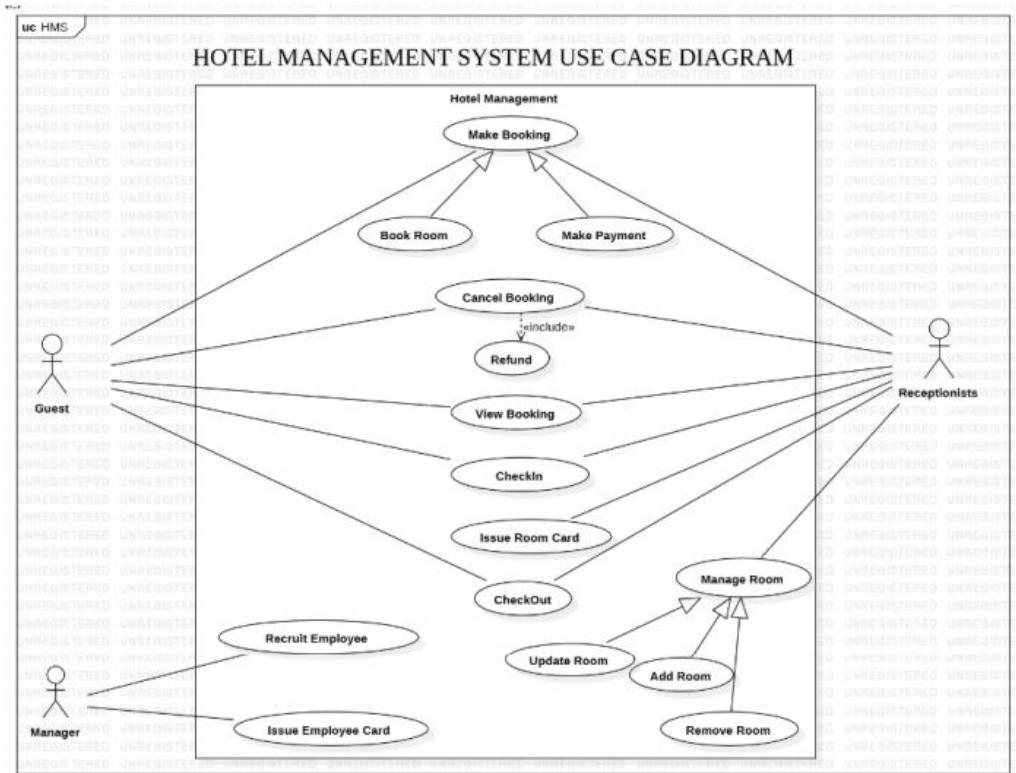


1.3 STATE DIAGRAM

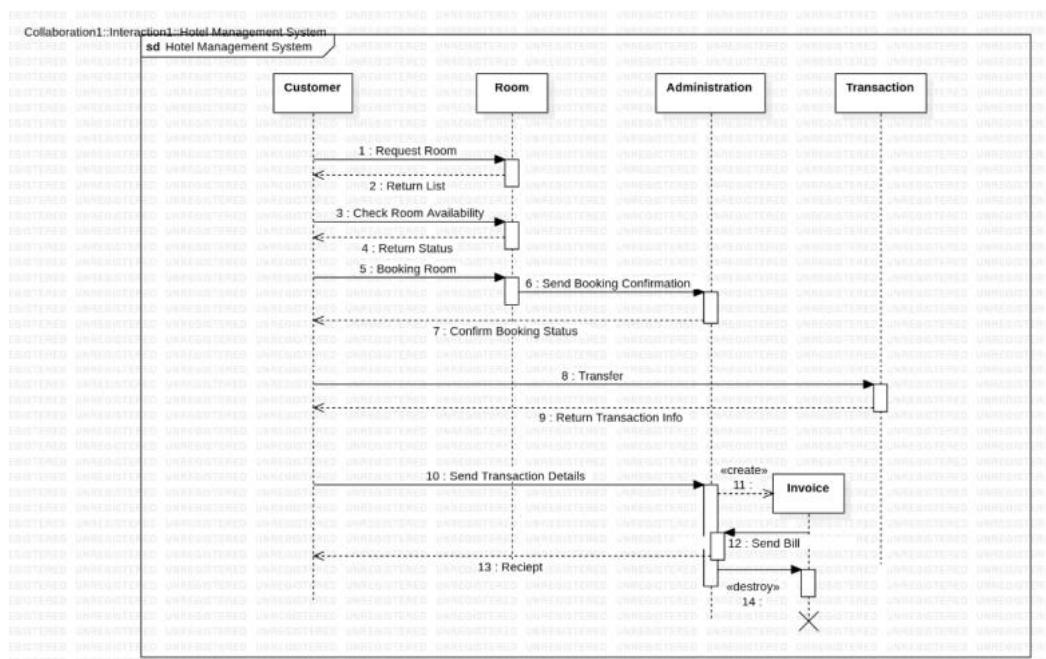


1.4 INTERACTION MODELS

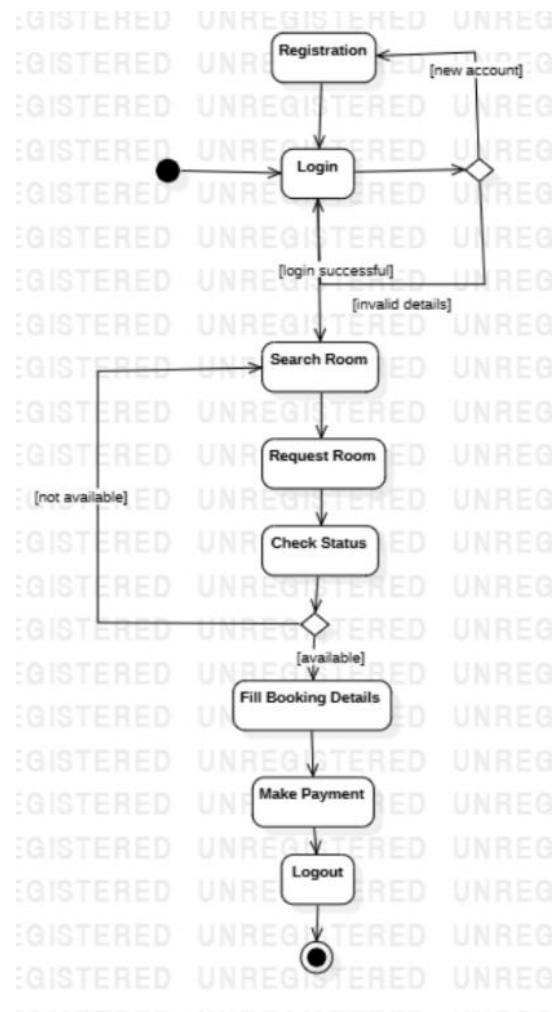
a. Use case model



b. Sequence model



c. Activity model



2.Credit Card Processing

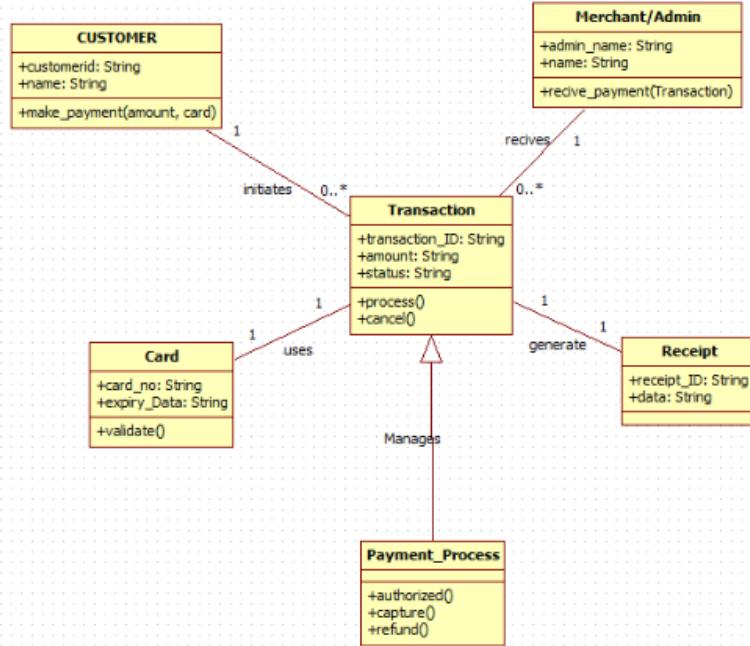
2.1 SRS DOCUMENT

SRS for Application 2	
Date	1/1/2024
Page	1/1
Credit Card Processing System	Version 1.0
1. Introduction	1.1 purpose
This document specifies the requirements for a Credit Card Processing System (CCPS) to securely handle online and in-person payment, ensuring compliance with security standard and efficient transaction processing.	
1.2 Documentation Convention	Functional Requirement are denoted as "FR", non-functional requirement as "NFR" and Extended interface requirement as "EIR".
1.3 Intended Audience and Reading Suggestions	This SRS is intended for developers, system architects, project managers and stakeholders involved in the CCPS implementation. It's recommended to start with "Overall description".
1.4 Project Scope	The CCPS will manage the process of authorizing, validating and completing credit card transaction. It will ensure that all transactions are secure, compliant with PCI-DSS.

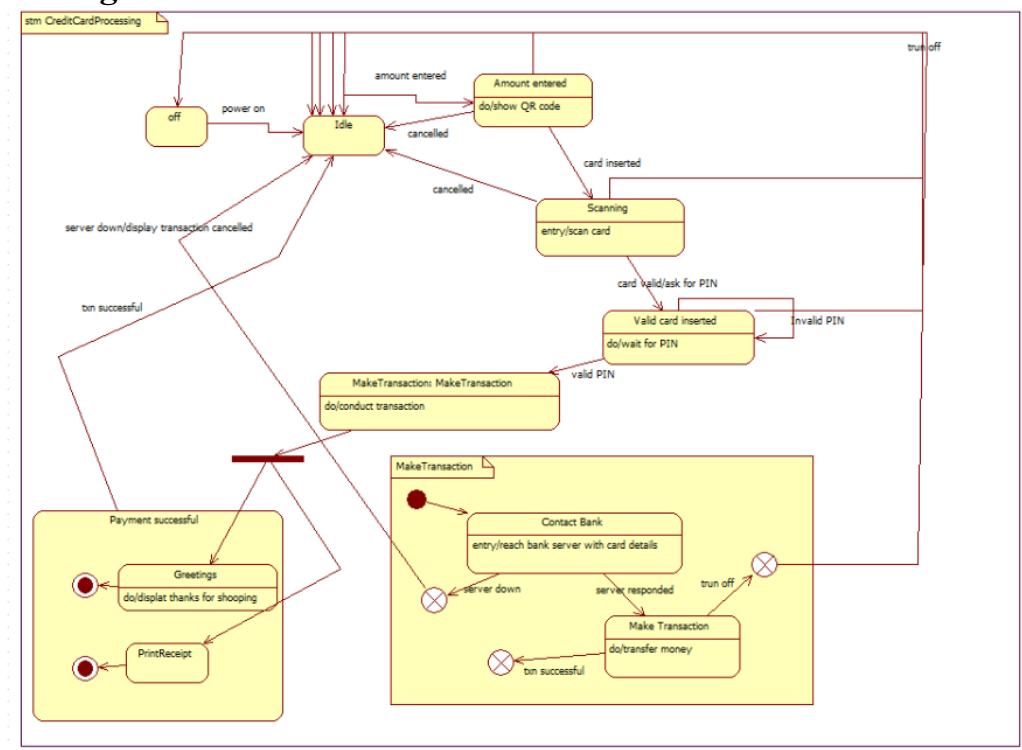
<p>References</p> <p>PCI-DSS ISO 27001 (compliant with PCI-DSS standard)</p> <p>2 Overall Description</p> <p>Product Perspective: The system will interact with payment gateways, banks, merchants and card issuers to facilitate credit card transactions. It will have a focus on security and compliance, ensuring sensitive data is encrypted.</p> <p>Product Functions:</p> <ul style="list-style-type: none"> → Transaction Authorization → Transaction Settlement → Fraud Detection → Payment Confirmation <p>User Characteristics:</p> <p>Merchants: Businesses integrating the API into their website / physical store.</p> <p>Bank: Financial institutions verifying & processing card transaction.</p> <p>Cardholders: Payment using their credit card.</p> <p>Advisors: Full access.</p>	<p>Constraints: Compliant with PCI-DSS standard. The system must process transaction with a 2 sec and it must see high volume transaction.</p> <p>Assumptions and Dependencies: The system assumes the availability of reliable internet. It depends on banks & payment gateway to validate and settle payment requests from merchants and cardholders.</p> <p>User Interfaced: Merchant Interface: A secure web portal for merchant to view transaction details.</p> <p>Administrator Interface: A dashboard for monitoring transaction.</p> <p>Cardholder Interface: Typically handled by merchant's website or POS system.</p> <p>APIs for integrating third-party systems.</p> <p>3 Specific Requirements:</p> <p>Functional requirements:</p> <ol style="list-style-type: none"> 1. FR1: The system must allow merchants to process credit card payment by sending authorization request to the cardholder's bank. 2. FR2: The system must support multiple payment methods. 3. FR3: The system must securely store transaction data, encrypted and tokenized, for later retrieval and dispute resolution. 4. FR4: The system must automatically flag suspicious transactions using predefined fraud detection rules.
---	---

<p>Non-Functional Requirements:</p> <ol style="list-style-type: none"> 1. NFR1: The system must process each transaction within 2 sec of receiving the authorization request. 2. NFR2: The system must be able to scale and handle up to 10000 transaction per minute during peak usage periods. 3. NFR3: The system must comply with PCI-DSS ensuring that sensitive credit card data is securely handled at all times. <p>Extended Interface Requirements:</p> <ol style="list-style-type: none"> 1. EIR 1: The system must integrate with various payment gateways, such as: <ul style="list-style-type: none"> EIR 1: The system must provide an API for third-party developers to integrate with merchant website or physical point-of-sale systems. EIR 3: The system should offer webhook notifications for merchants on transaction status update. 	<p>Appendices:</p> <p>A1: UML diagram for system architecture, including the payment flow.</p> <p>A2: Flowcharts illustrating the payment processing cycle, from authorization to settlement.</p>
--	--

2.3 Class Diagram

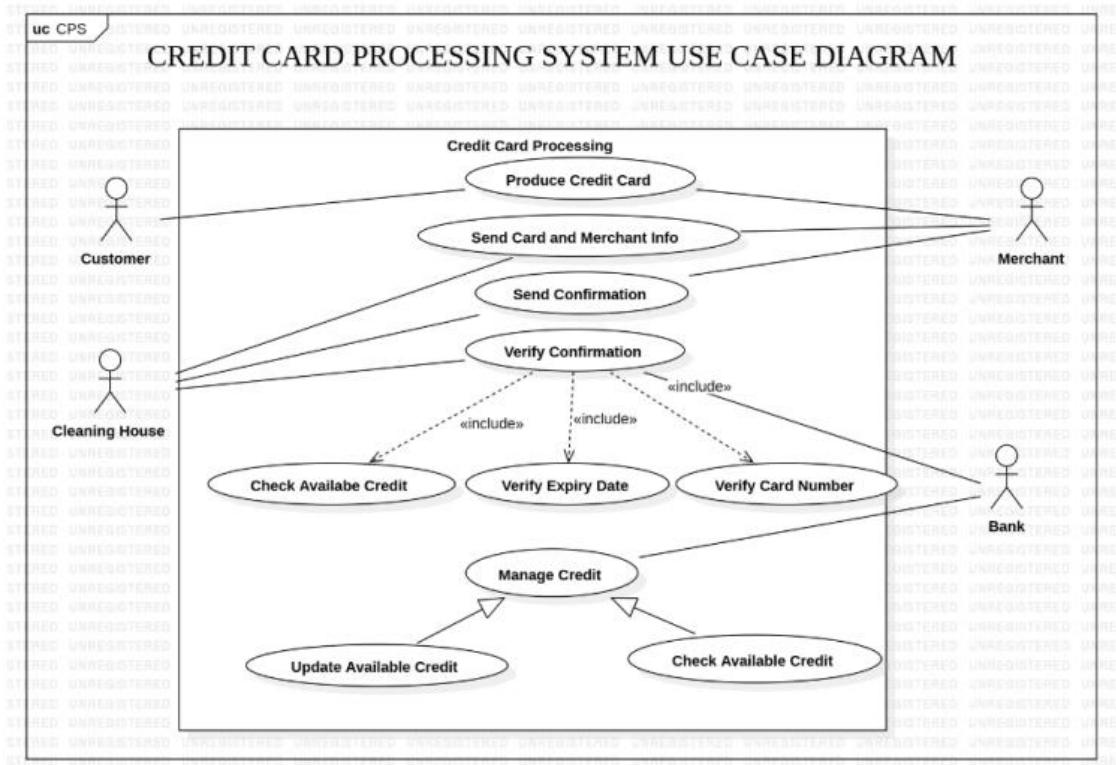


2.4 State Diagram

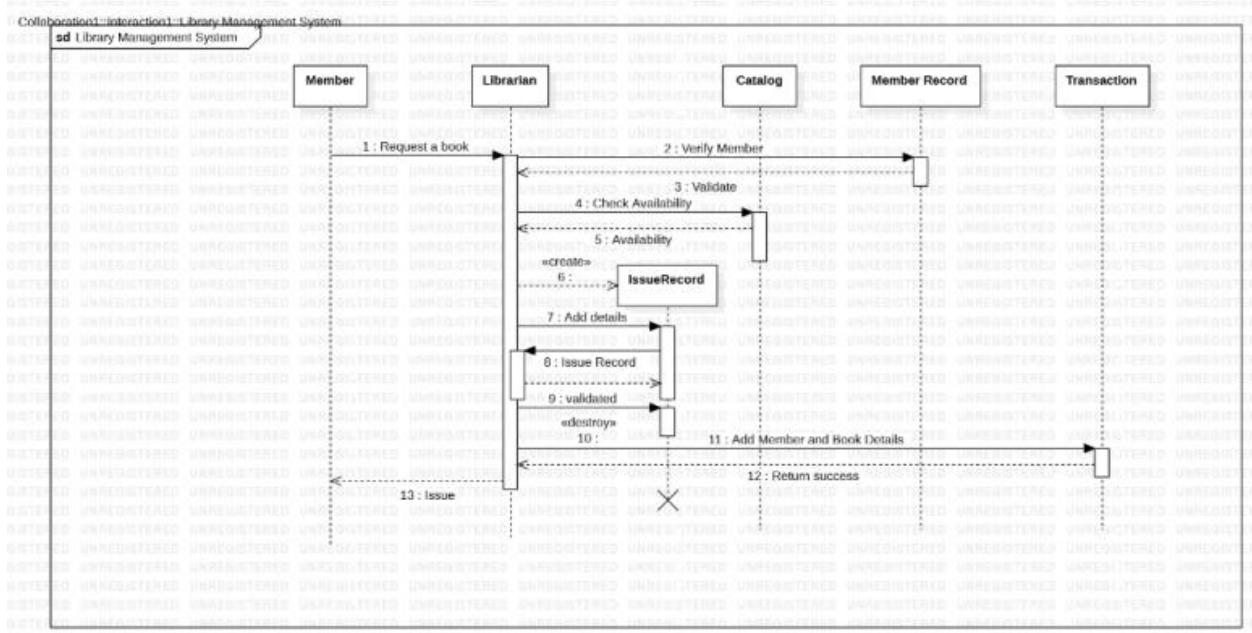


2.5 Interaction Models

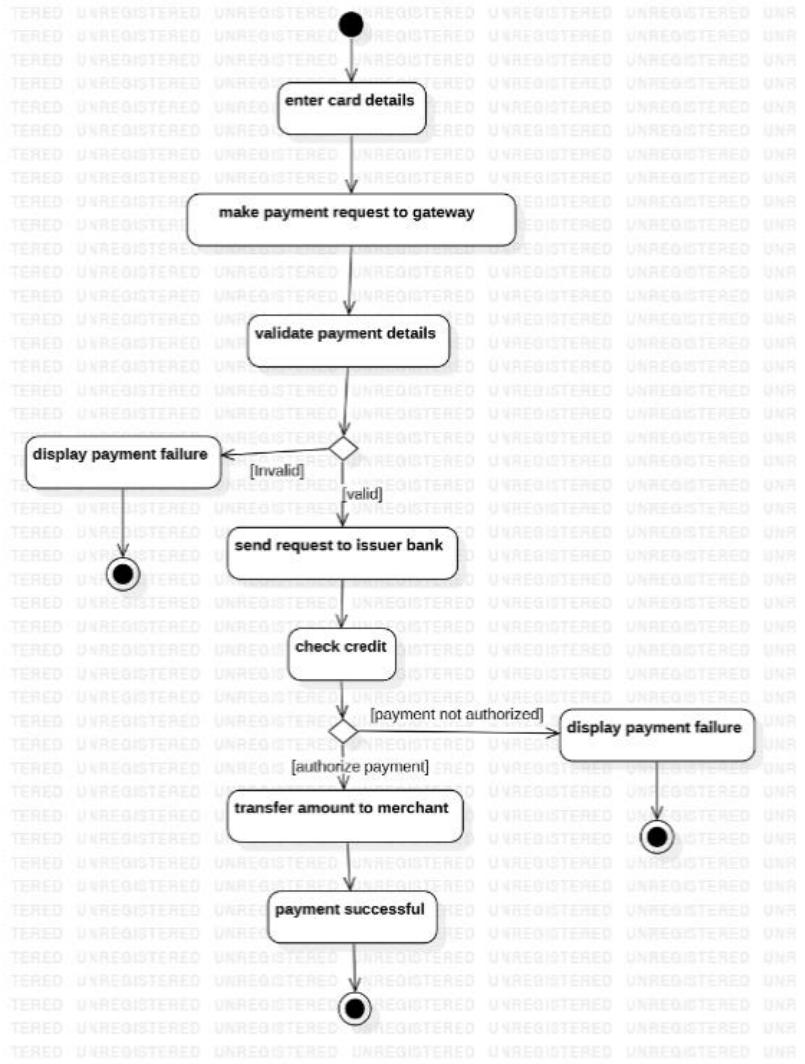
a. Use Case Model



d. Sequence model



c. Activity Model



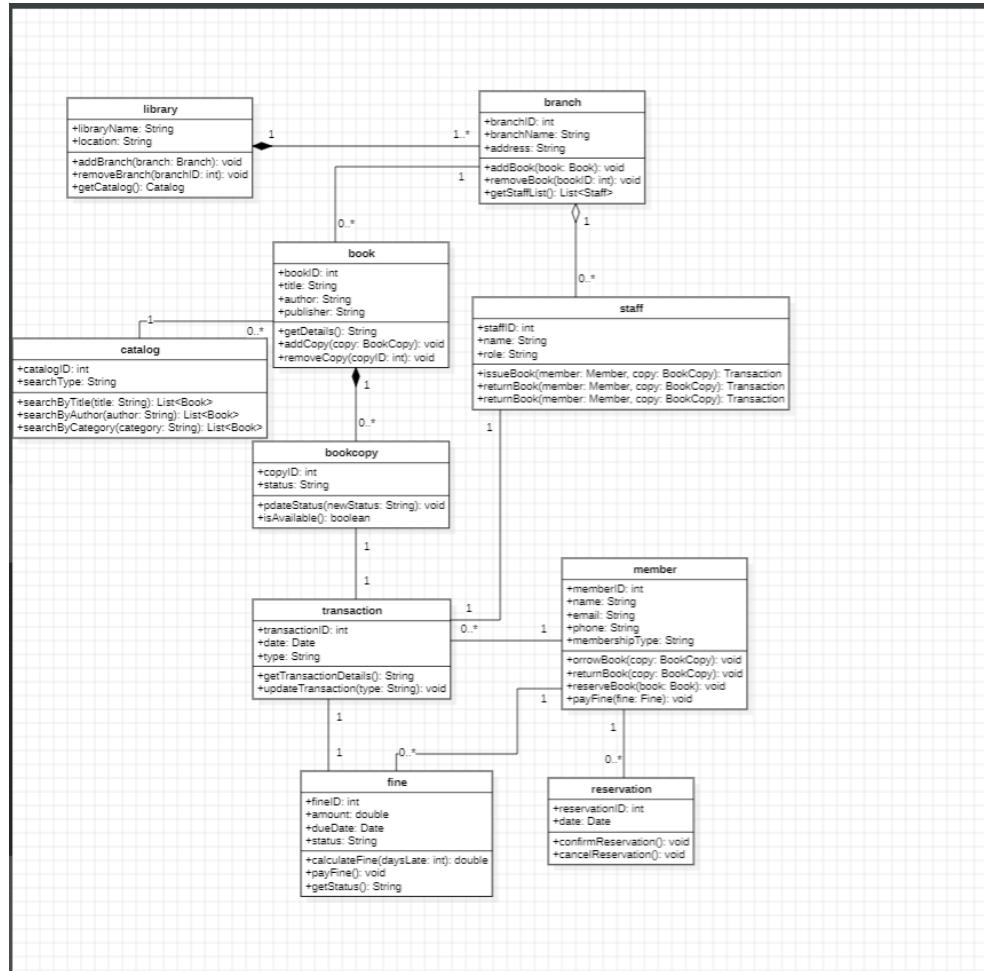
3. Library management system

3.1 SRS Document

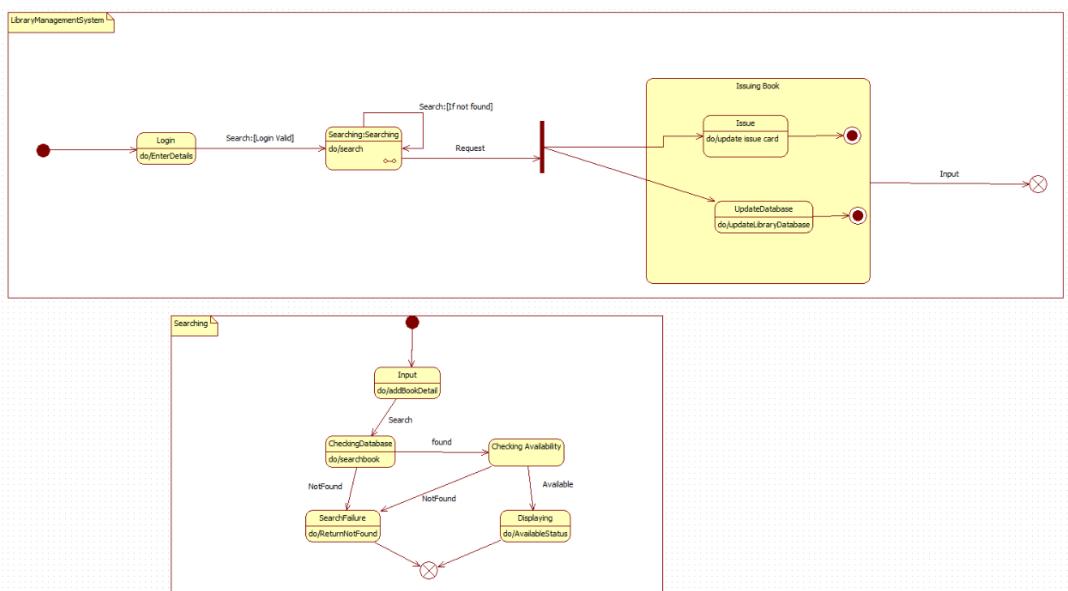
LAB - 2		Date / / Page / /
Application : 03		
Library Management System		
Problem Statement		
Traditional library systems depend on manual book keeping, which leads to inefficiency, misplaced records, and difficulties in tracking borrowed or returned books.		1.3 Intended Audience and Reading Suggestion The document is meant for developers, librarians, system administrators and students.
A Library Management system [LMS] is required to automate book management, borrowing, returns, and user registrations to improve efficiency and accessibility.		1.4 Project Scope The LMS manages books, users, borrowing and return operations. It will help librarians organize books efficiently.
SRS		1.5 References : IEEE 820-1998 SRS standard.
Introduction:		2 Overall Description
1.1 purpose The purpose of this system is to automate library operations including book issues, return, catalog management, and users records.		2.1 product perspective The system will act as a central database application for storing library records connected to a user-friendly interface for both staff and students.
1.2 Document Convention: This SRS follows IEEE 820-1998 standard. Requirements are classified into functional.		2.2 product Functions: The LMS will provide functions such as book search, borrowing, return, catalog update & fine calculation.
		2.3 User classes and characteristics: • Librarians: Manage books • Students/Users: Search and borrow books • Administrators: System monitoring and reports
		2.4 Operating Environment: The system will run on desktops with database support and may extend to web or mobile access.

Date / / Page / /		Date / / Page / /
2.5 Design and Implementation Constraints		3.3. Non-functional Req.
The system must be use a relational database. Authentication and role-based access are required.		→ Reliability → System must not lose record during update.
2.6 User Documentation		→ Availability → Accessible 24 hours (3 AM to 10 PM) for online access.
user manuals and quick guide will be provided for staff and students.		→ Security → user authentication and role-based access controls.
2.7 Assumptions & Dependencies		3.2 product Functions → Managing books and membership records. Generates reports and provides alerts when new user is registered.
It is assumed that a stable internet or local network connection will be available.		3.4 User Classes : existing SIS → Existing student information system.
3 Specific Requirements		3.5 Operating Environment → Database management system (MySQL, PostgreSQL, Oracle, etc.) → Web server (Apache, Nginx, IIS, etc.) → Application server (Tomcat, JBoss, GlassFish, etc.) → Database management system (MySQL, PostgreSQL, Oracle, etc.) → Web server (Apache, Nginx, IIS, etc.) → Application server (Tomcat, JBoss, GlassFish, etc.)
3.1 Functional Requirements		
→ Add, update, and delete book records		
→ Borrow and return books with due dates		
→ Calculate and apply fines for late returns		
→ Generate library usage reports		
3.2 External Interface Requirements		
→ User Interfaces → Logins Screen, book search, form, borrowing / return Panel.		
→ Hardware Interface → SQL Database Student record sys.		
→ Communication Interface : LAN, internet for remote access.		

3.3 Class Model

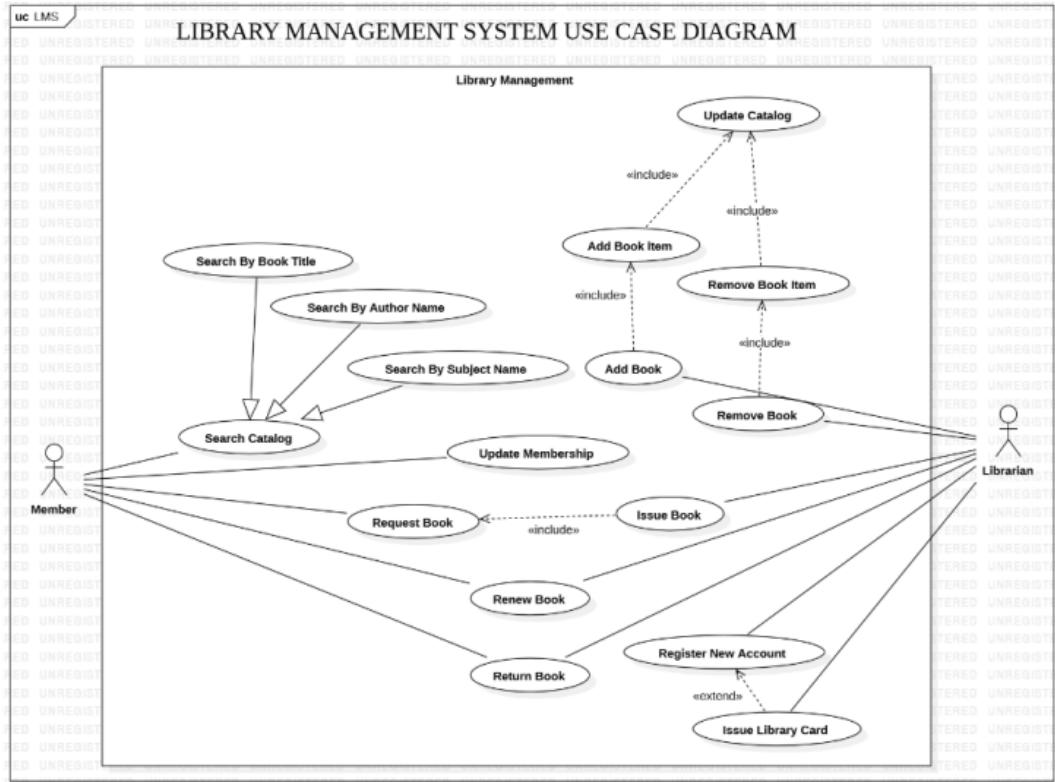


3.4 State model

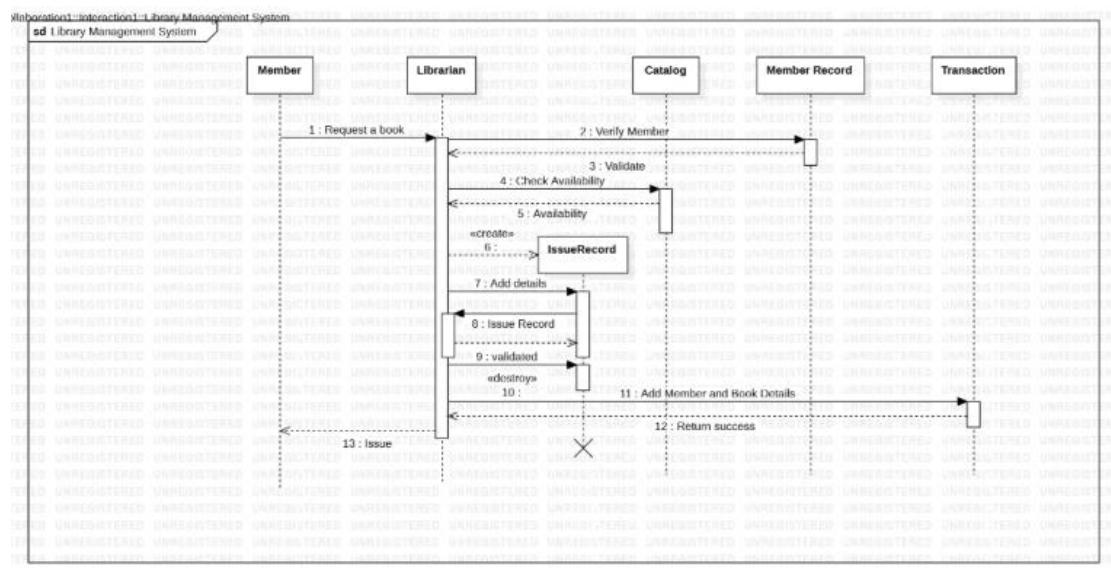


3.5 Interaction Models

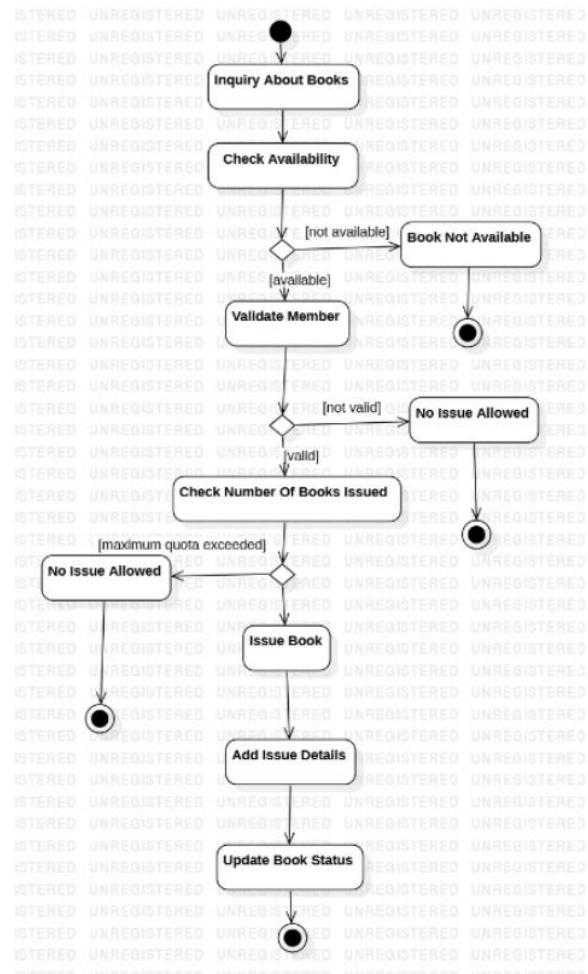
a. Use Case Model



b. Sequence Model



c. Activity Model



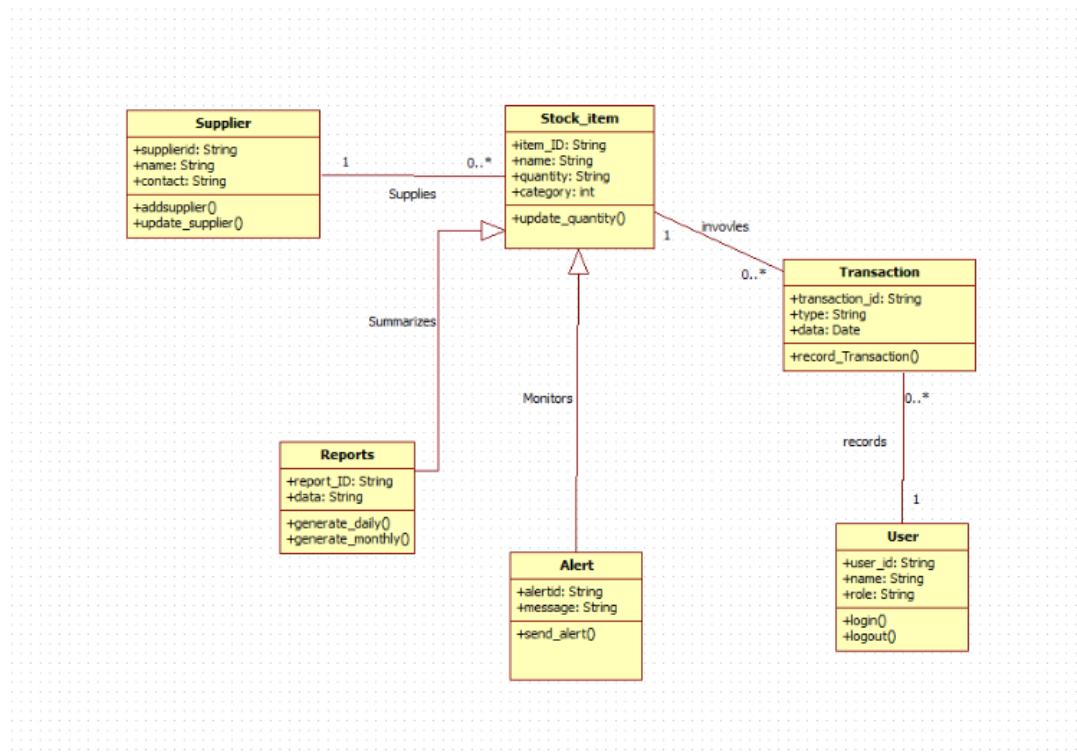
4. Stock Maintenance System

4.1 SRS Document

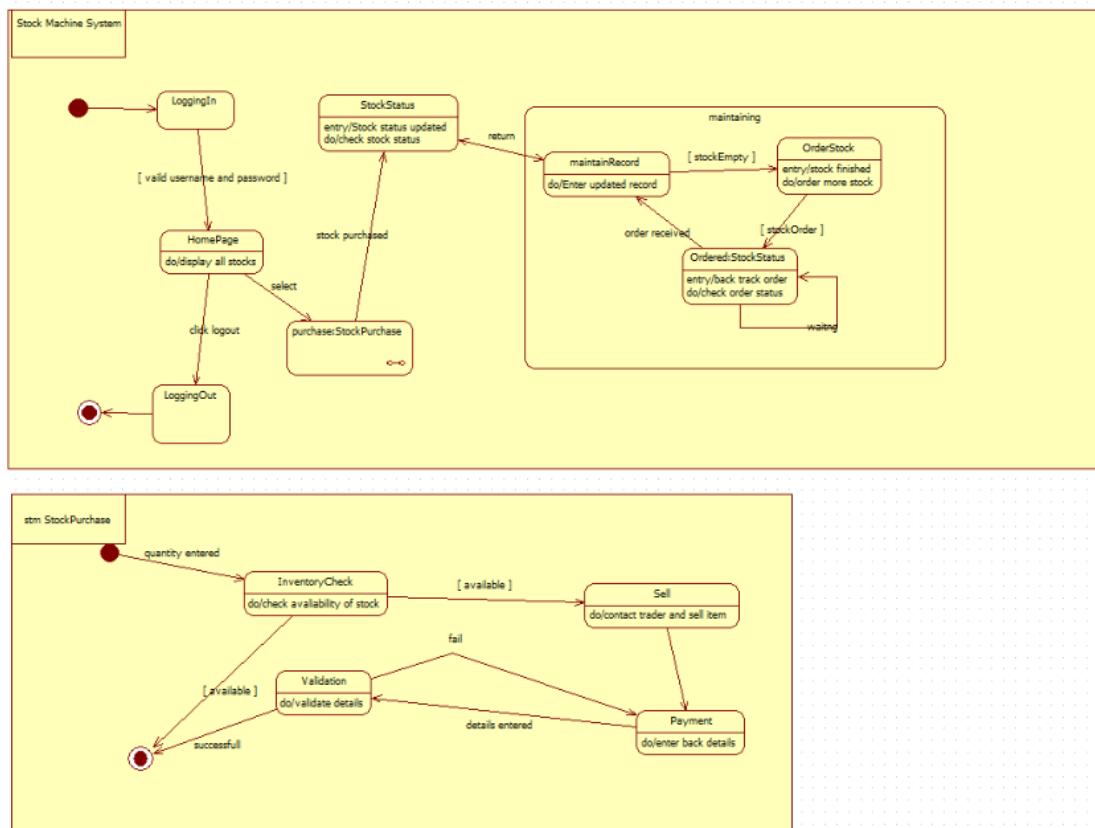
Date _____ Page _____	Date _____ Page _____
Application - 04 Stock Maintenance System:	1.1 project Scope The system manages stock-in, stock-out, supplier details, and alerts for low inventory.
Problem Statement: Manual stock management often leads to errors, misplaced records and difficulties in tracking inventory levels. A stock maintenance system is needed to automate stock-in, stock-out, and reporting process, ensuring accuracy and efficiency in inventory control.	1.5 References: IEEE 830-1998 Standard
Software Requirements Specification (SRS)	2 Overall Description 2.1 product perspective The stock maintenance system is a stand-alone application with a database to track and manage inventory.
1. Introduction: 1.1 purpose: The purpose of the system is to automate stock records, track item availability and generate reports for efficient inventory control.	2.2 product Functions: The system maintains stock records, generates reports and provides alerts when items are running low.
1.2 Document Conventions: This document follows IEEE 830-1998 SRS guidelines.	2.3 user classes: • Store Manager → oversees stock • Staff Members → update stock entries • Administrator → Manages system settings
1.3 Intended audience: For use by store managers, warehouse staff and developers.	2.4 Operating Environment: Runs on desktop / mobile with support programming language.
	2.5 constraints: Required secure login, a stable database, and compliance with data accuracy.

Date _____ Page _____	Date _____ Page _____
2.6 User Documentation: User manual & quick-start guide will be provided.	→ maintainability is easy to add new items/categories and reports.
2.7 Assumptions & Dependencies: Assumes continuous internet connection and functional hardware.	→ maintainable maintainability
3 Specific Requirements: 3.1 Functional Requirements: → Add, update, and delete stock items → Record stock-in and stock-out transaction → Generate daily / weekly / monthly reports → Send alerts for low stock levels	→ maintainable maintainability
3.2 External Interface Req: → User Interface → Dashboard for stock → Hardware → Desktop / mobile device → barcode Scanner → Software → SQL database integration → Communication → LAN / Internet	→ maintainable maintainability
3.3 Non-functional Requirements: → Reliability → System must ensure data accuracy & consistency. → Security → only authorized users can modify stock records → performance → stock update should process within 2 seconds.	→ maintainable maintainability
	3.4 Testing: testing with different test cases → Test cases: 1.1 basic functionality test → Test cases: 2.2 security test → Test cases: 3.3 performance test

4.3 Class model

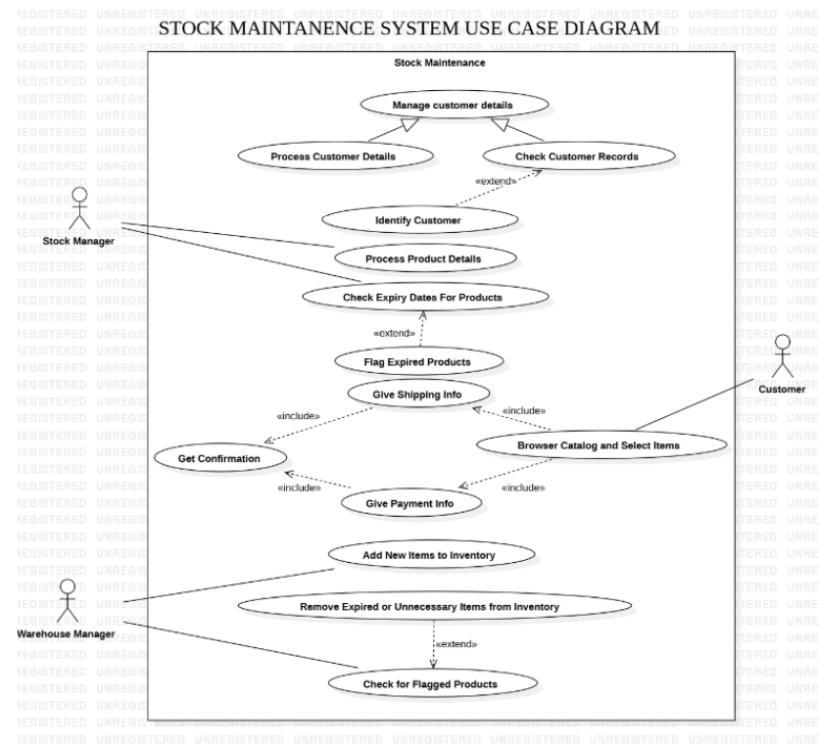


4.4 State Model

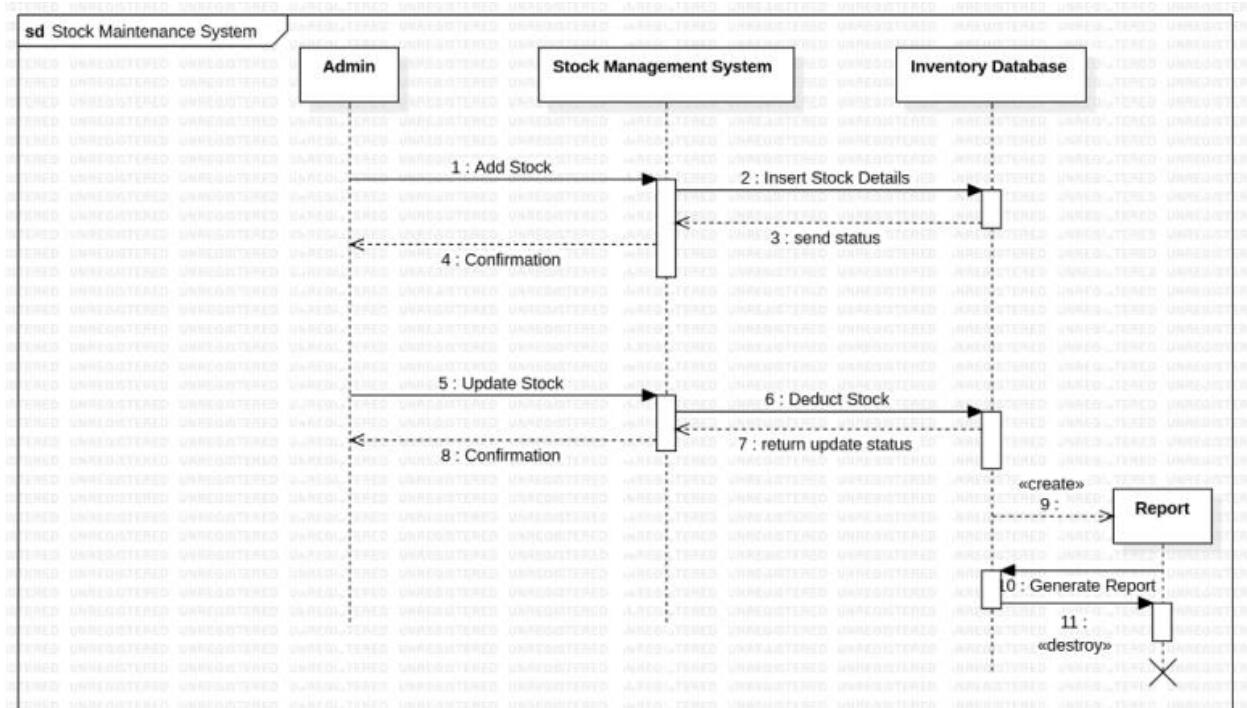


4.5 Interaction Models

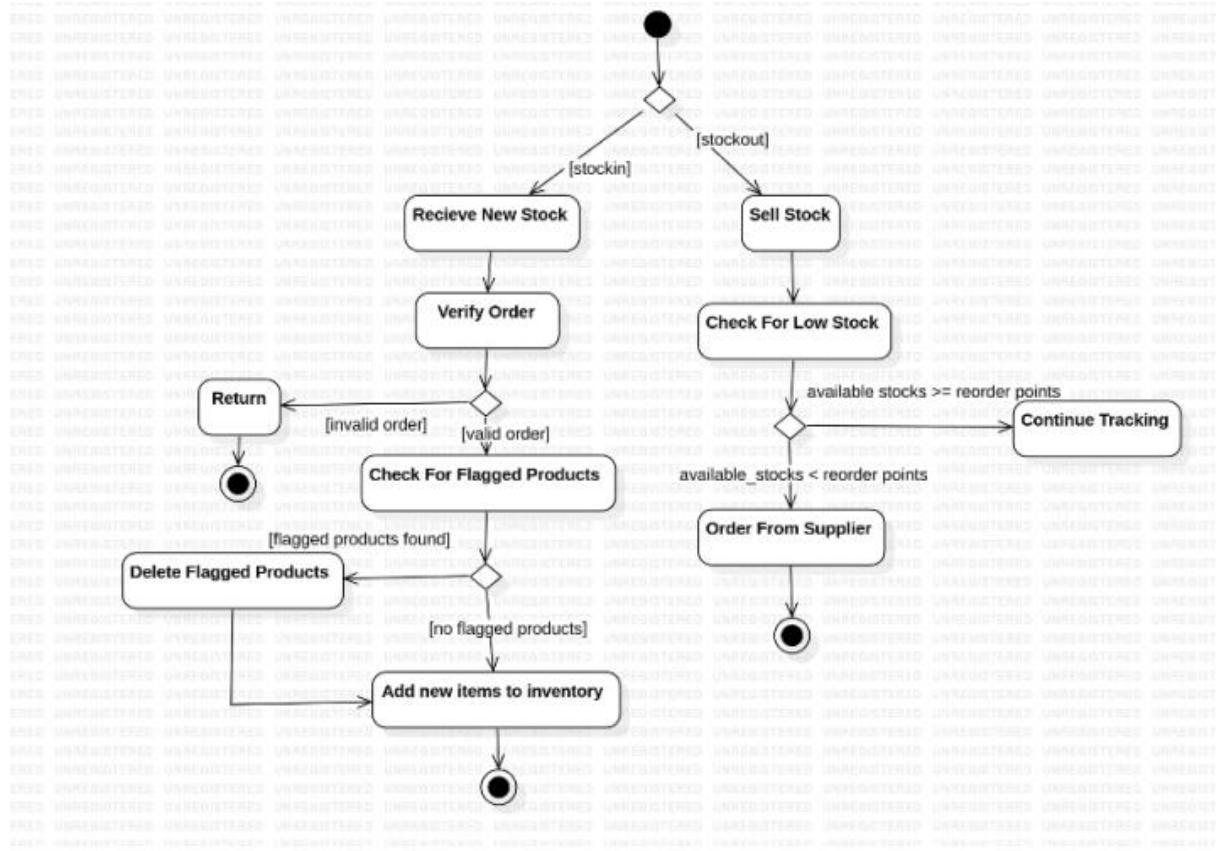
a. Use Case Model



d. Sequence Model



e. Activity Diagram



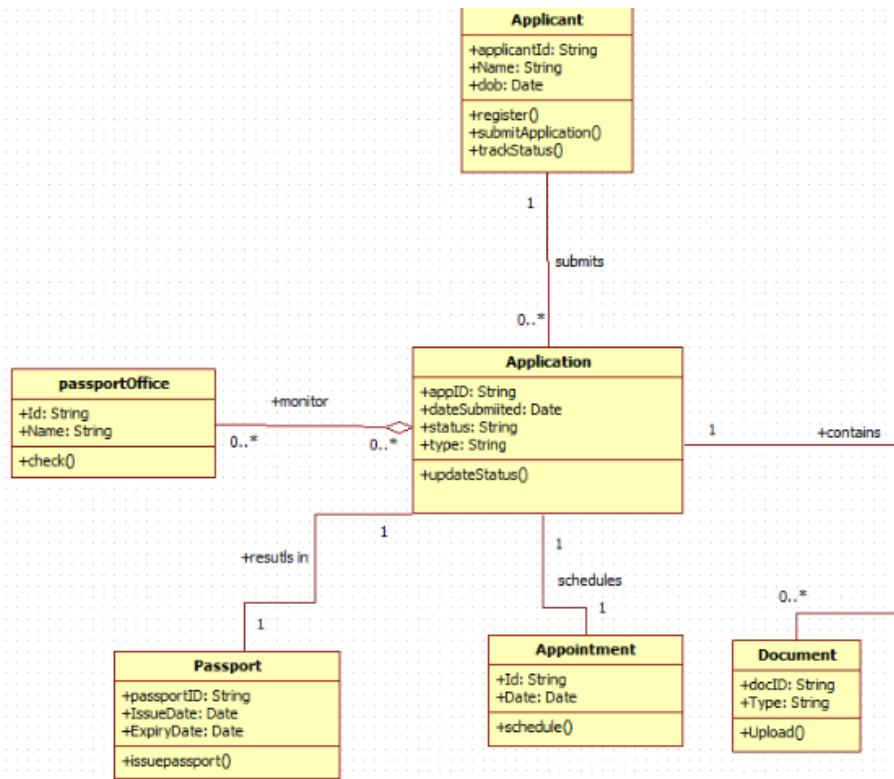
5 .Passport Automation System

5.1 SRS Document:

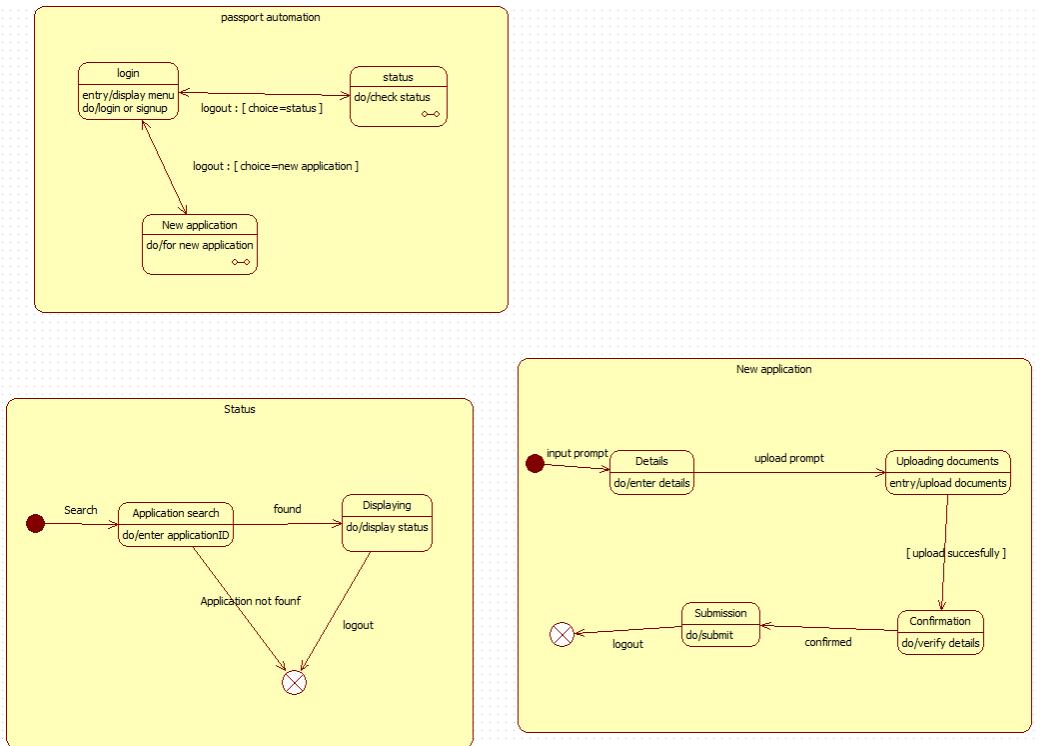
Application - 05		Date / /	Page / /
	passport Automation system's		
	problem statement:		
	The manual process of passport application and verification often result in delays, errors and lack of transparency. A passport Automation System is req to automate application Submission, verification, and tracking ensuring faster processing and improved accuracy.		
	<u>SYS</u>		
1.	1. Introduction		
1.1	purpose :		
	The passport Automation system aims to automate the process of applying, verifying and issuing passports. It will reduce manual errors and improve efficiency.		
1.2	Document Conventions		
	This document follows IEEE 830-1998 standard for software requirement specifications		
1.3	Intended Audience		
	This document is intended for applicants, passport officials, developers, system administrators.		
1.4	project Scope		
	The sys allows online passport application, verification of details, appointment scheduling and status tracking		
1.5	References		
	IEEE 830-1998 standard		
2.	Overall Description		
2.1	product Perspective		
	The system will act as an online platform integrated with a gov database for passport processing.		
2.2	product Functions		
	It will handle new applications, renewals, verification, appointments, and issue status updates		
2.3	User Classes		
	→ Applicant → Applies for passport and tracks status		
	→ passport officer → verifies applications and documents		
	→ Administrator → Manages the system and reports		
2.4	Operating Environment		
	The system will run as a web-based application with database connectivity and internet access.		

Requirement : 8-8A		Date / /	Page / /
2.5	Non Functional Req		
	Requires Secure authentication compliance with gov policies and reliable servers		
2.6	User Documentation		
	User manual & online help will be provided for applicants and staff		
2.7	Assumptions & Dependencies		
	Depends on gov ID verification		
3.	Specific Req		
3.1	Functional Req		
	→ Allows applicants to register & submit applications online		
	→ Provide doc upload & verification feature		
	→ Enable appointment scheduling for verification / interviews		
	→ Enable status tracking for applicants		
3.2	External Interface Req		
	→ User Interface → Online forms for application and status tracking		
	→ Hardware → Desktop / mobile access, biometric Scanner		
	→ Software → Database, gov verification APIs		
	→ Communication → Secure intermodule connection		
4.	Appendix		
	PASS → Passport Automation Sys		
	SS2 → Secure Socket Layer		

5.2 Class Model

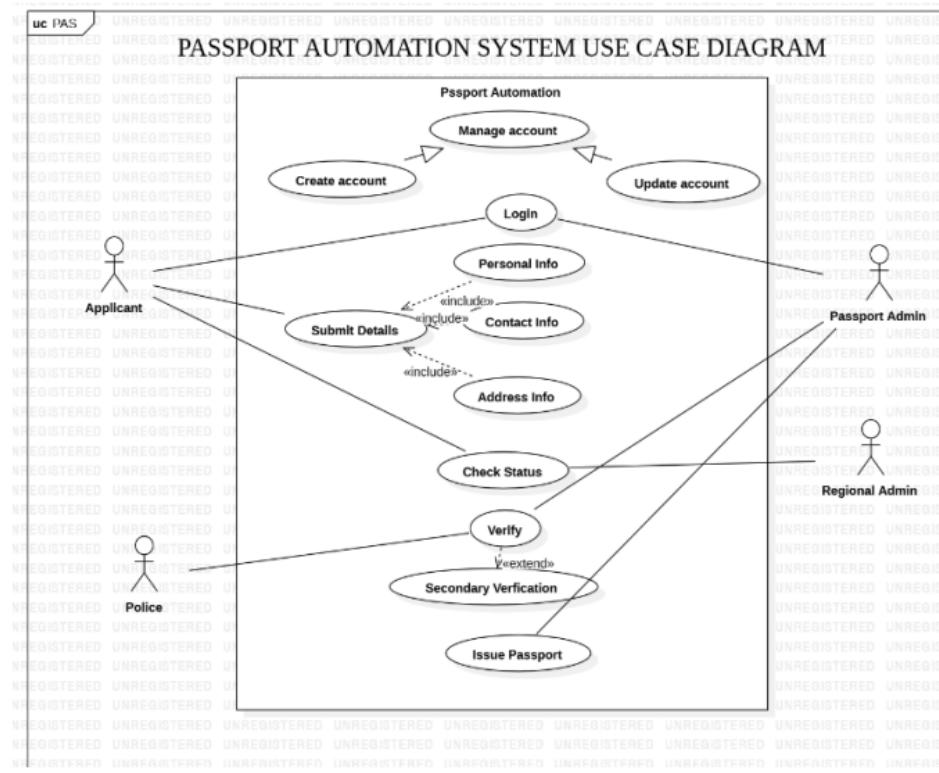


5.3 State Model

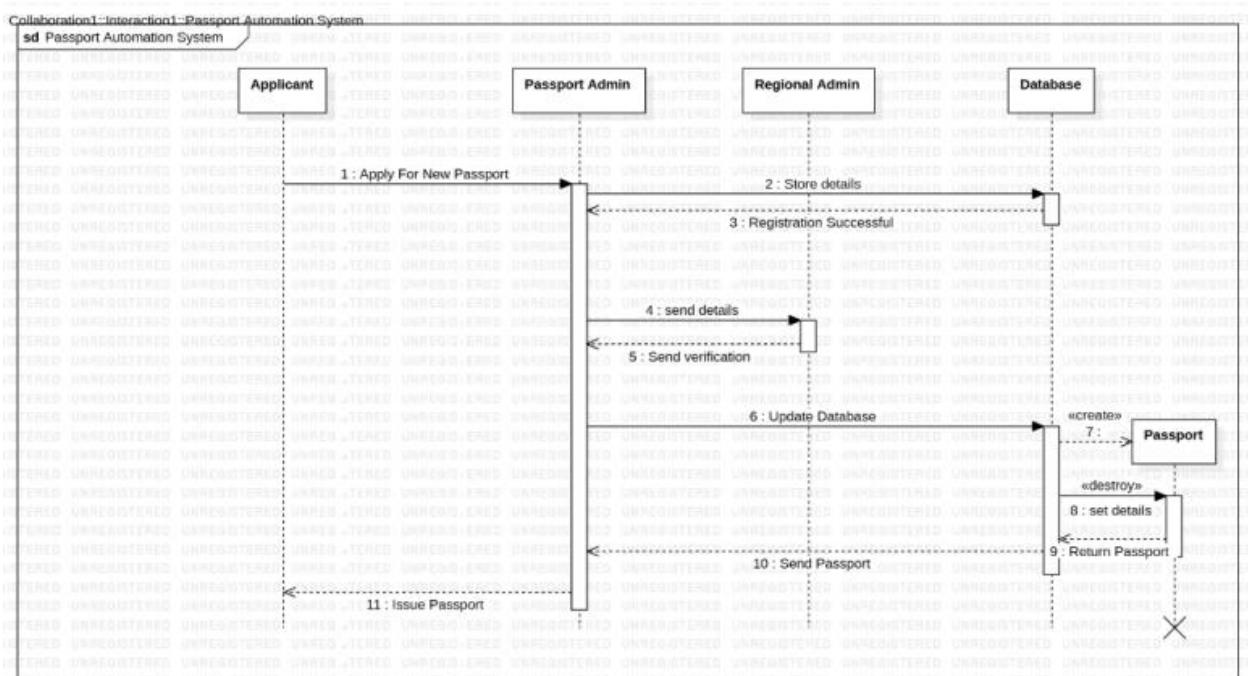


5.4 Interaction Models

a. Use Case Model



a. Sequence Model



b. Activity model

