VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

Object Oriented Modelling

(23CS5PCOOM)

Submitted by:

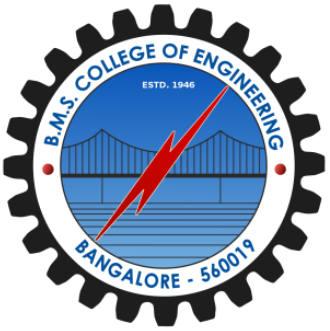
***GHANSHYAM SHARMA*(1BM23CS100)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

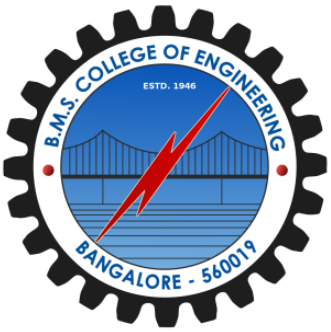
(Autonomous Institution under VTU)

BENGALURU-560019

February 2025 - June 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 Modelling**" carried out by **GHANSHYAM SHARMA (1BM23CS100),** who is 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 2025-26. The Lab report has been approved as it satisfies the academic requirements in respect of **Object Oriented Modelling - (23CS5PCOOM)** work prescribed for the said degree.

Sonika Sharma D
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Table Of Contents

Course Outcomes (COs):

CO1: Apply the knowledge of class, State & Interaction Modelling using Unified Modeling Language to solve a given problem

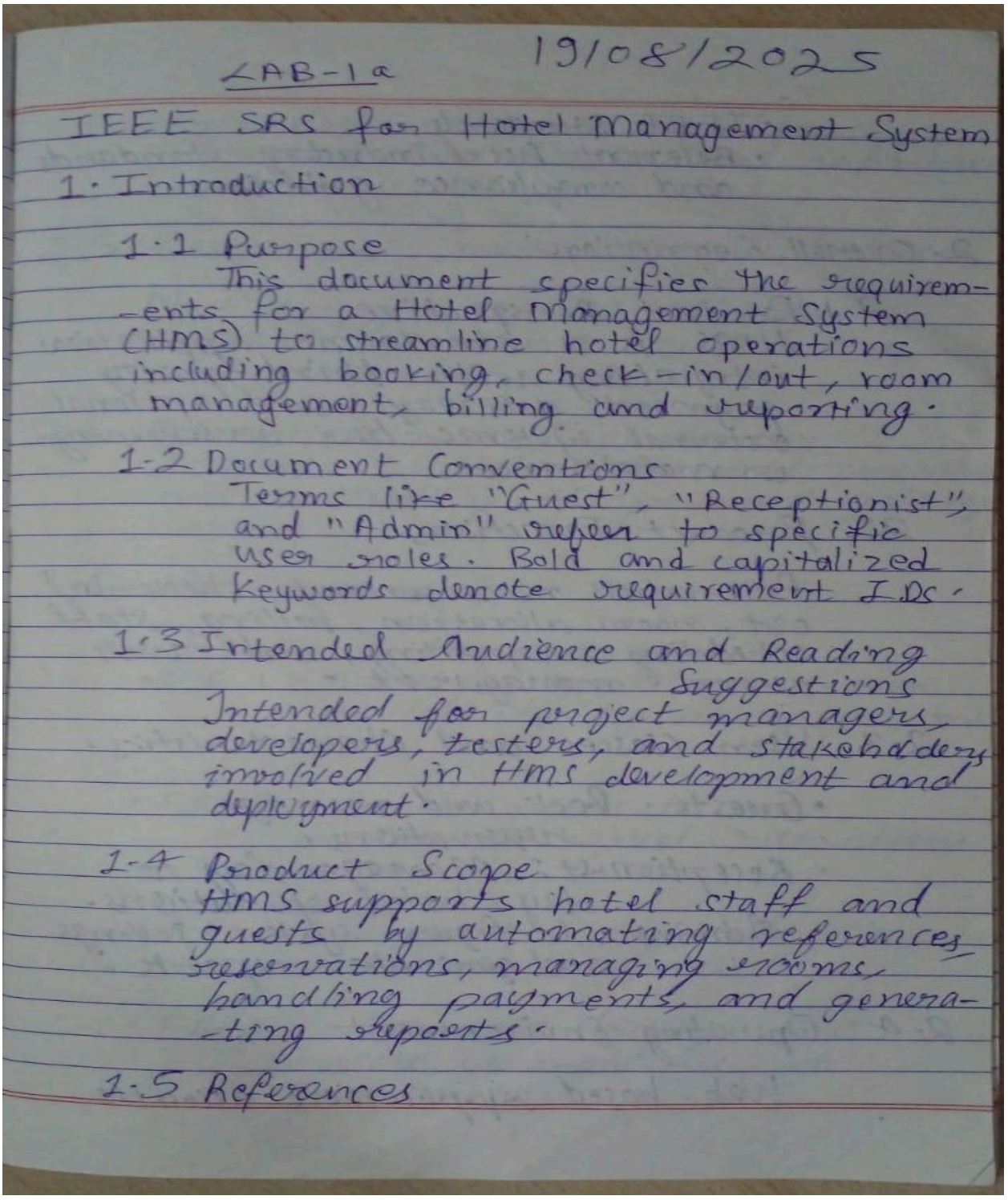
CO2: Analyze a System for a given requirement using Unified Modeling language

CO3: Design a given system using high level strategy

CO4: Conduct practical experiment to solve a given problem using Unified Modeling language.

Hotel management System

SRS



LAB-1a 19/08/2025

IEEE SRS for Hotel Management System

1. Introduction

1.1 Purpose

This document specifies the requirements for a Hotel Management System (HMS) to streamline hotel operations including booking, check-in/out, room management, billing and reporting.

1.2 Document Conventions

Terms like "Guest", "Receptionist", and "Admin" refer to specific user roles. Bold and capitalized Keywords denote requirement IDs.

1.3 Intended Audience and Reading Suggestions

Intended for project managers, developers, testers, and stakeholders involved in HMS development and deployment.

1.4 Product Scope

HMS supports hotel staff and guests by automating references reservations, managing rooms, handling payments, and generating reports.

1.5 References

- IEEE Standards
- Relevant hotel industry standards and compliance guidelines.

2. Overall Description

2.1 Product Perspective

HMS is a standalone application integrating with third-party payment gateways; and optional external systems like housekeeping or maintenance.

2.2 Product Functions.

Reservation management, check-in/out, room allocation, billing, staff scheduling, reporting, and user account management.

2.3 User Classes and Characteristics:

- Guests: Book and manage reservations.
- Receptionist: Manage day to day hotel operations.
- Admins: Configure system settings and generate reports.

2.4 Operating Environment

Web-based application accessible

via desktop and mobile browsers; backend hosted on cloud infrastructure.

2.5 Design and Implementation Constraints

Must comply with PCI-DSS for payment processing; support up to 500 concurrent users.

2.6 Assumptions and Dependencies

Assumes reliable internet connection and availability of payment gateway services.

3. Specific Requirements

3.1 Functional Requirements

- FR-1: Allow guests to search room availability by date, type, and price.
- FR-2: Enable guests to make, modify, or cancel reservations online.
- FR-3: Support receptionist check-in and room assignment.
- FR-4: Automate billing and invoice generations at check-out.

- FR-5 : Implement user authentication and role-based access control
- FR-6 :- Allow admins to generate booking, revenue, and occupancy reports.

3.2 External Interface Requirements

- User Interface : Responsive web UI with role-specific dashboards
- Hardware Interface : Optional card reader integeration for key ~~instance~~ issuance.
- S/w Interface : APIs for payment gateways and house-keeping systems.

3.3 Non Functional Requirement (NFRs)

- NFR-1 (Performance)
  Support 500 concurrent users; response time < 3 seconds.
- NFR-2 (Security)
  Encrypt sensitive data; multi-factor auth for admins.
- NFR-3 (Reliability)
  99.9% uptime with failover

- NFR-4 (Usability):
  Intuitive, WCAG 2.1 compliant interface.

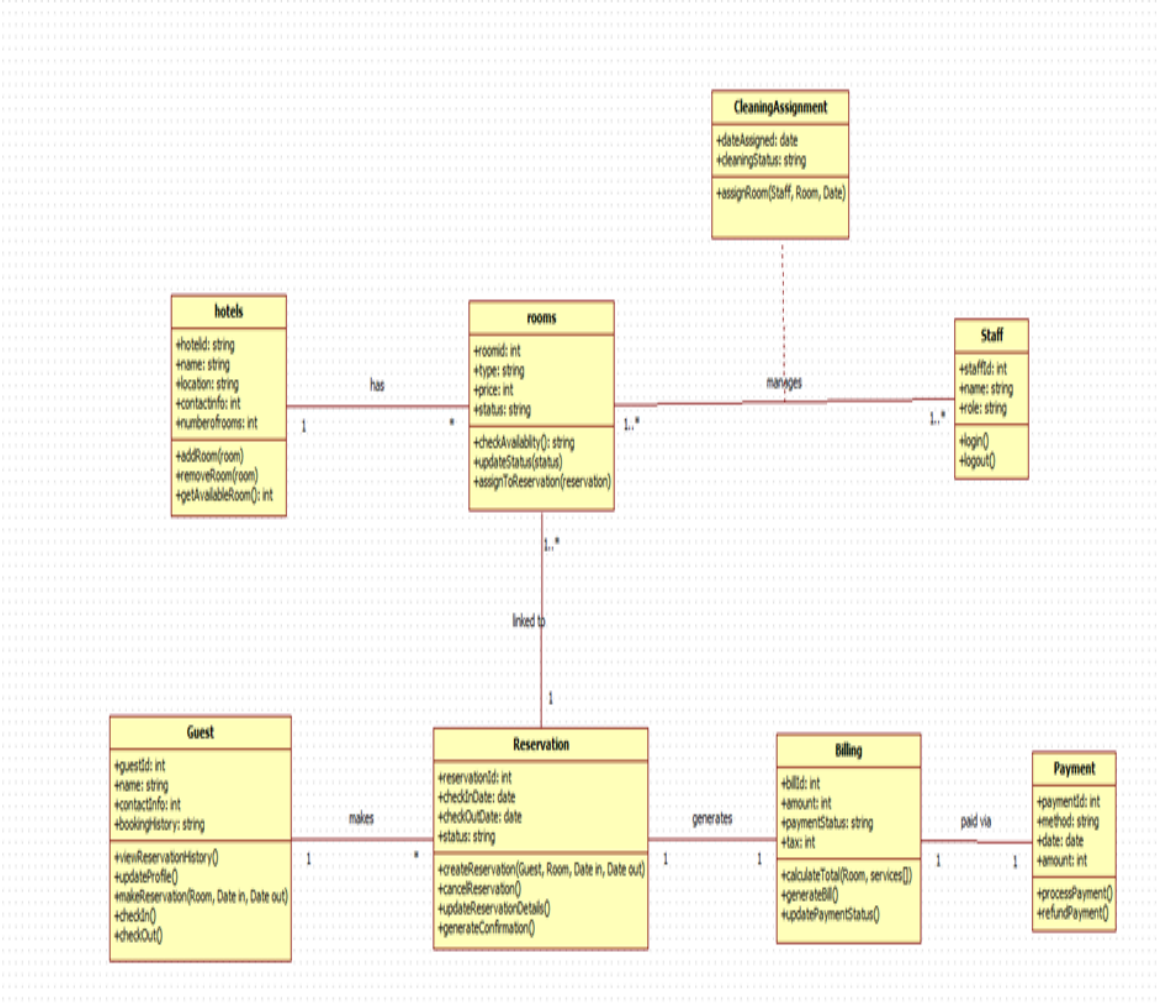
4. Appendices
   Additional info: glossary, user manuals, sample reports, database overview.

5. Glossary

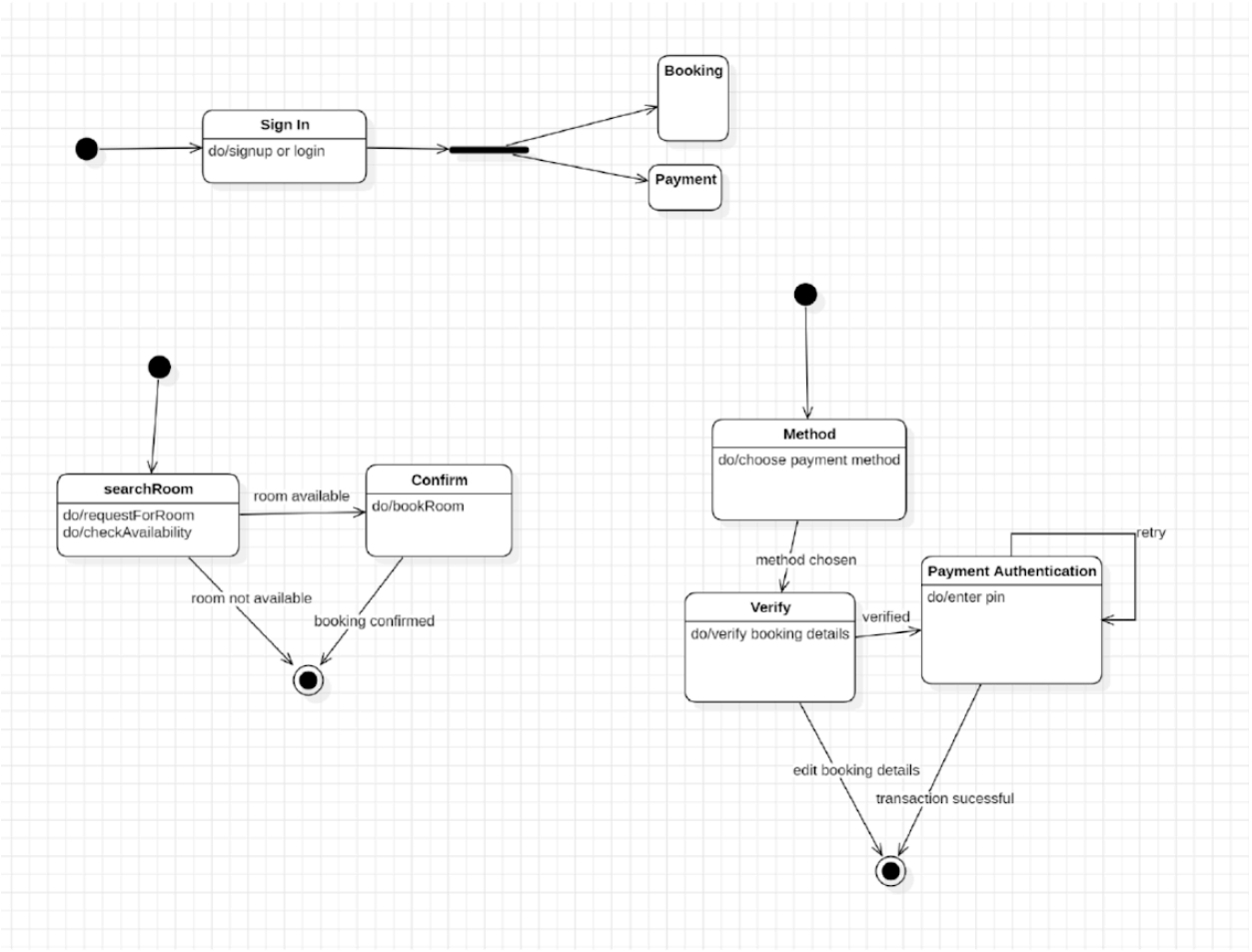
- Guest: Hotel customer
- Receptionist: Front desk staff
- Admin: System administrator
- Reservation: Room booking
- PCI-DSS: Payment security standard.

NL
19/8/25

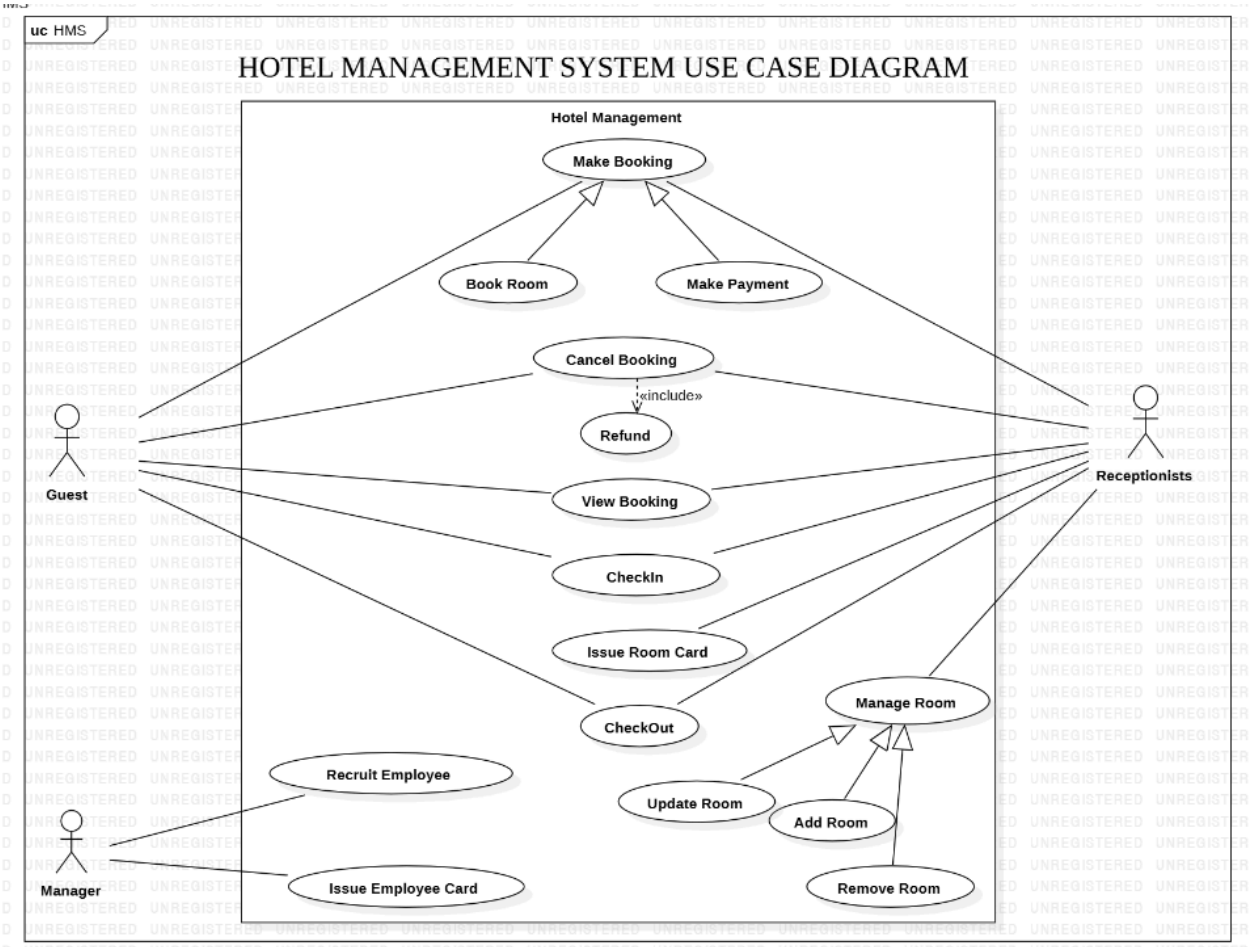
Class Diagram



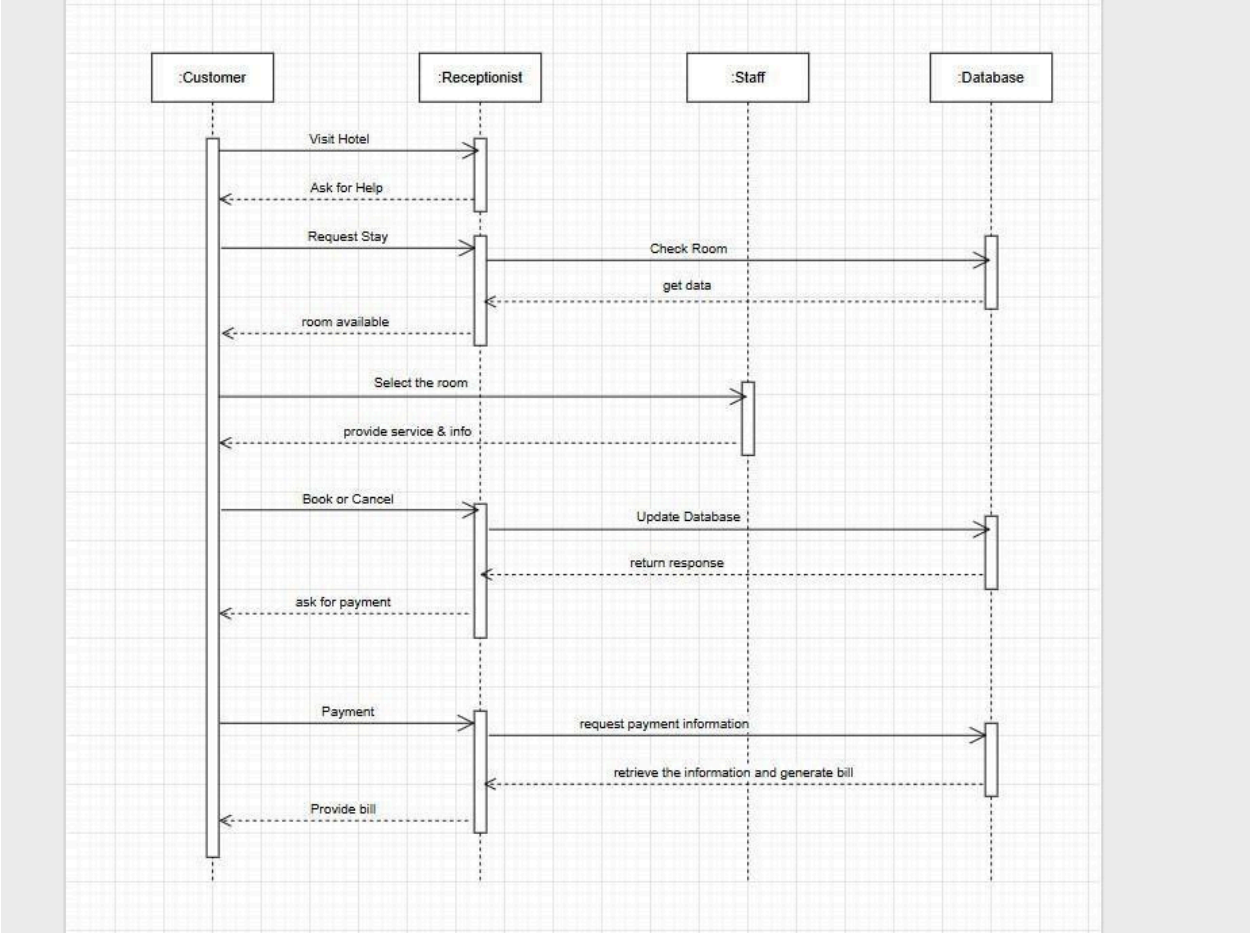
State Diagram



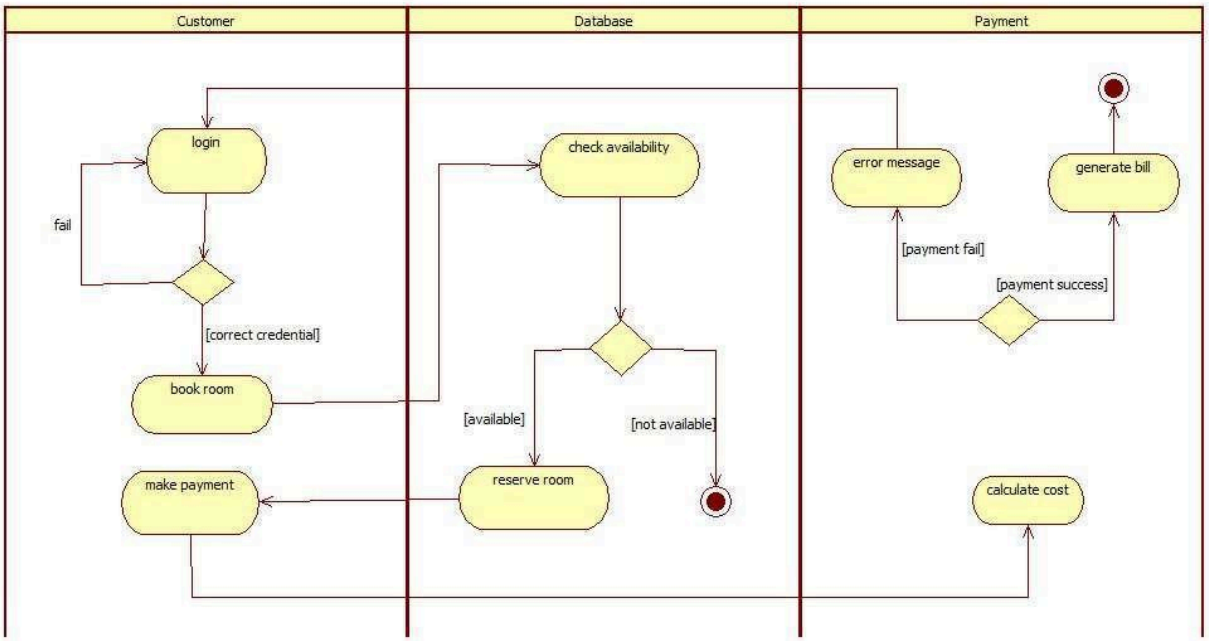
Use Case Diagram



Sequence Diagram

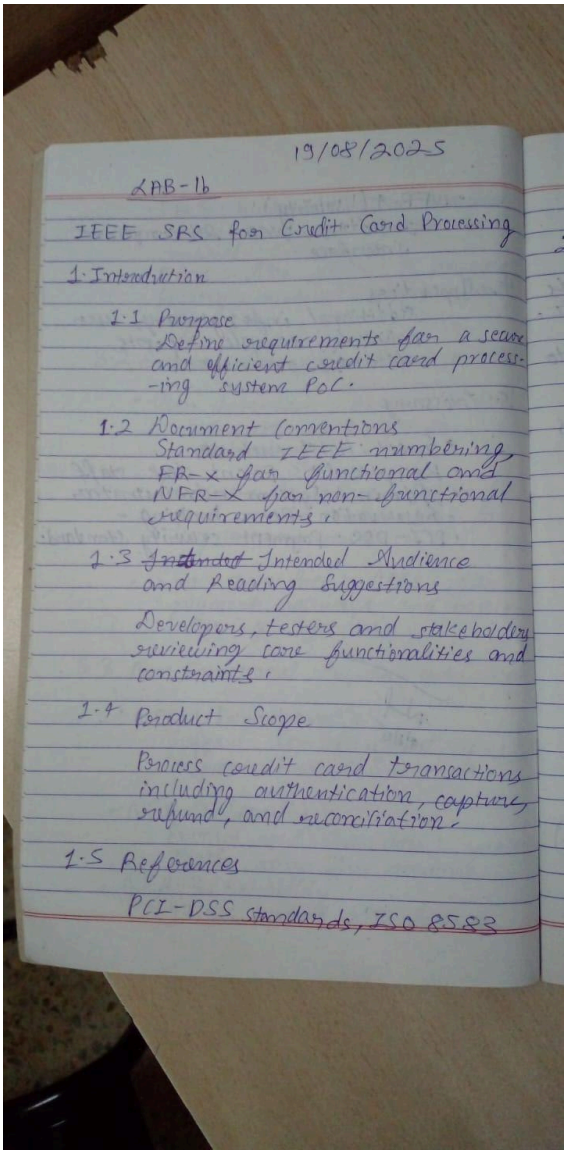


Activity Diagram



Credit Card Processing

SRS



19/08/2025

ZAB-16

IEEE SRS for Credit Card Processing

1. Introduction

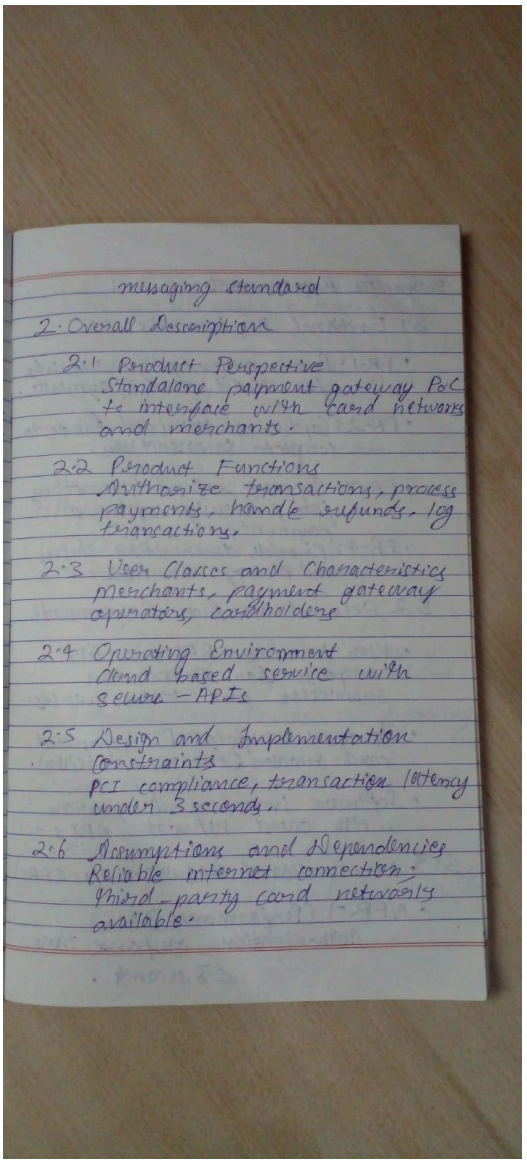
1.1 Purpose
Define requirements for a secure and efficient credit card processing system PoC.

1.2 Document Conventions
Standard IEEE numbering, FR-x for functional and NFR-x for non-functional requirements.

1.3 ~~Intended~~ Intended Audience and Reading Suggestions
Developers, testers and stakeholders reviewing core functionalities and constraints.

1.4 Product Scope
Process credit card transactions including authentication, capture, refund, and reconciliation.

1.5 References
PCI-DSS standards, ISO 8583



messaging standard

2. Overall Description

2.1 Product Perspective
Standalone payment gateway PoC to interface with card networks and merchants.

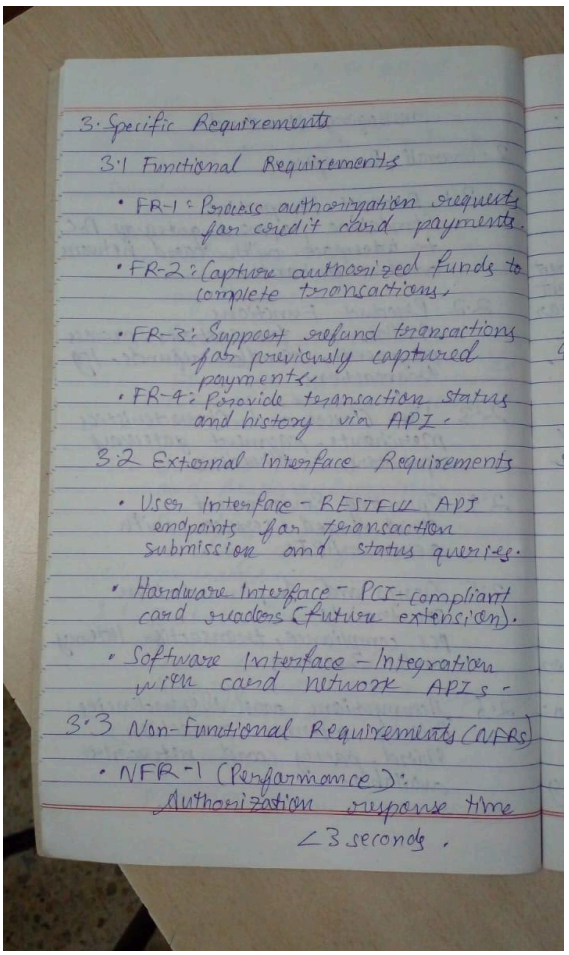
2.2 Product Functions
Authorize transactions, process payments, handle refunds, log transactions.

2.3 User Classes and Characteristics
merchants, payment gateway operators, cardholders

2.4 Operating Environment
cloud based service with secure - APIs

2.5 Design and Implementation constraints
PCI compliance, transaction latency under 3 seconds.

2.6 Assumptions and Dependencies
Reliable internet connection. Third-party card networks available.



3. Specific Requirements

3.1 Functional Requirements

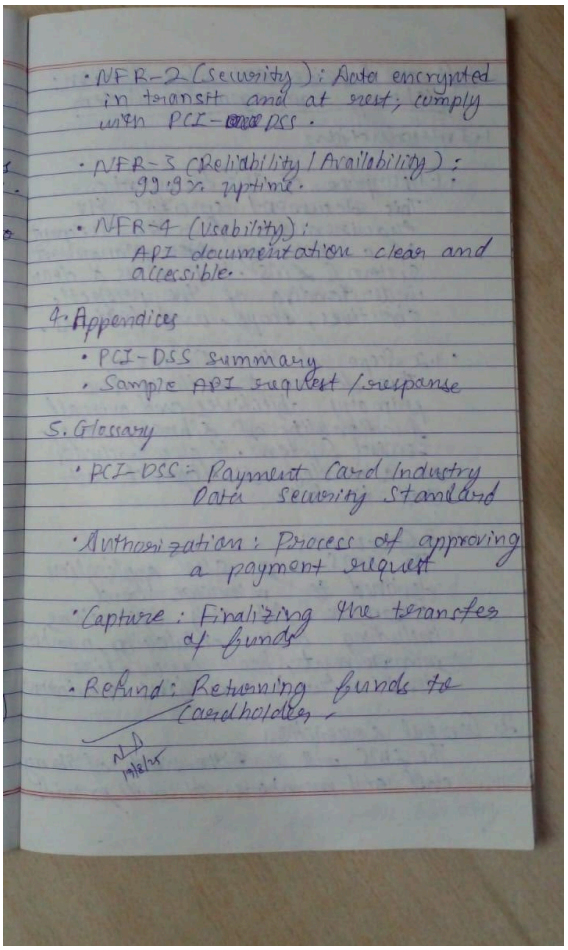
- FR-1: Process authorization requests for credit card payments.
- FR-2: Capture authorized funds to complete transactions.
- FR-3: Support refund transactions for previously captured payments.
- FR-4: Provide transaction status and history via API.

3.2 External Interface Requirements

- User Interface - RESTful API endpoints for transaction submission and status queries.
- Hardware Interface - PCI-compliant card readers (future extension).
- Software Interface - Integration with card network APIs.

3.3 Non-Functional Requirements (NFRs)

- NFR-1 (Performance): Authorization response time < 3 seconds.



- NFR-2 (Security): Data encrypted in transit and at rest; comply with PCI-DSS.
- NFR-3 (Reliability / Availability): 99.9% uptime.
- NFR-4 (Usability): API documentation clear and accessible.

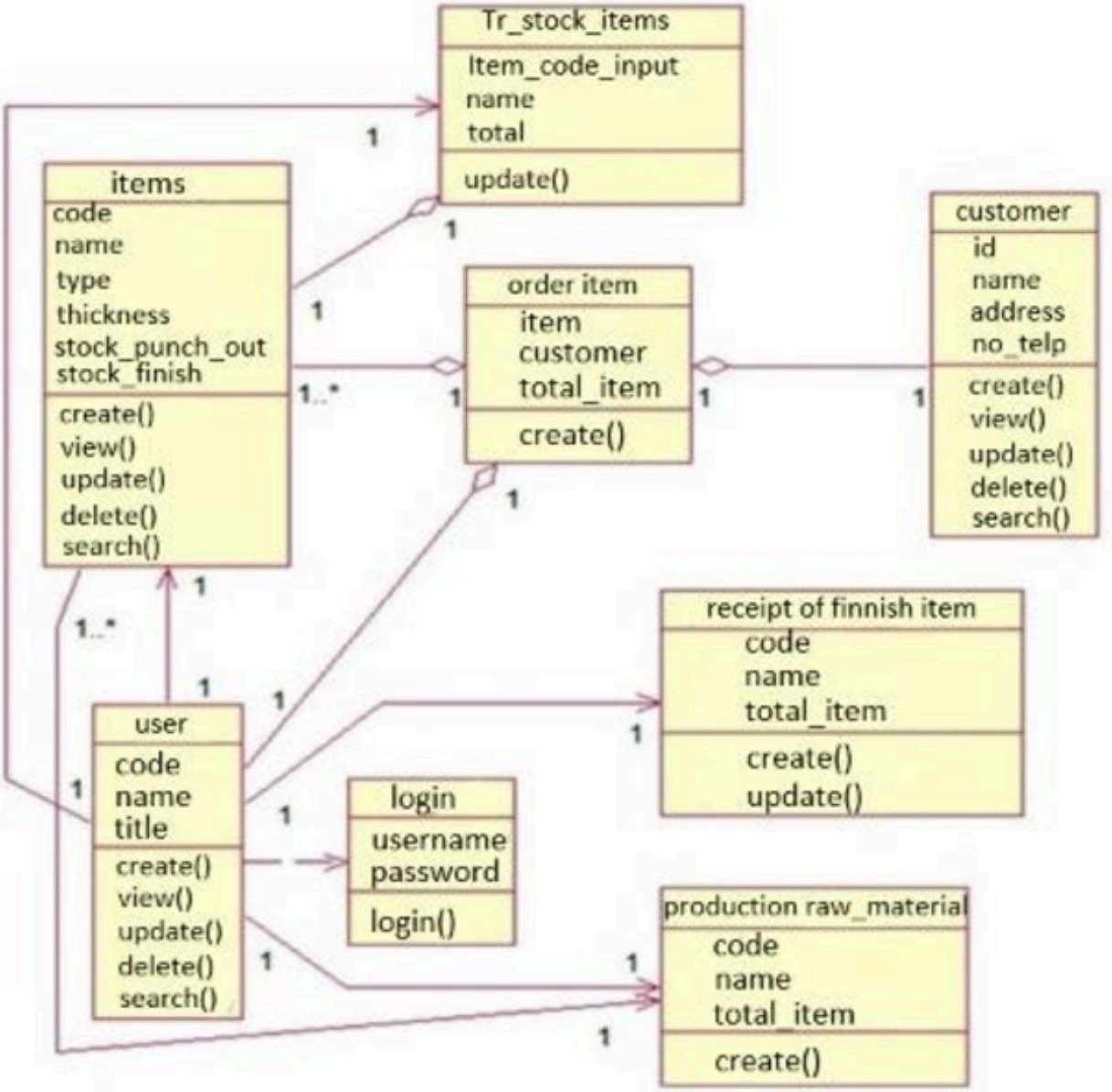
4. Appendices

- PCI-DSS summary
- Sample API request / response

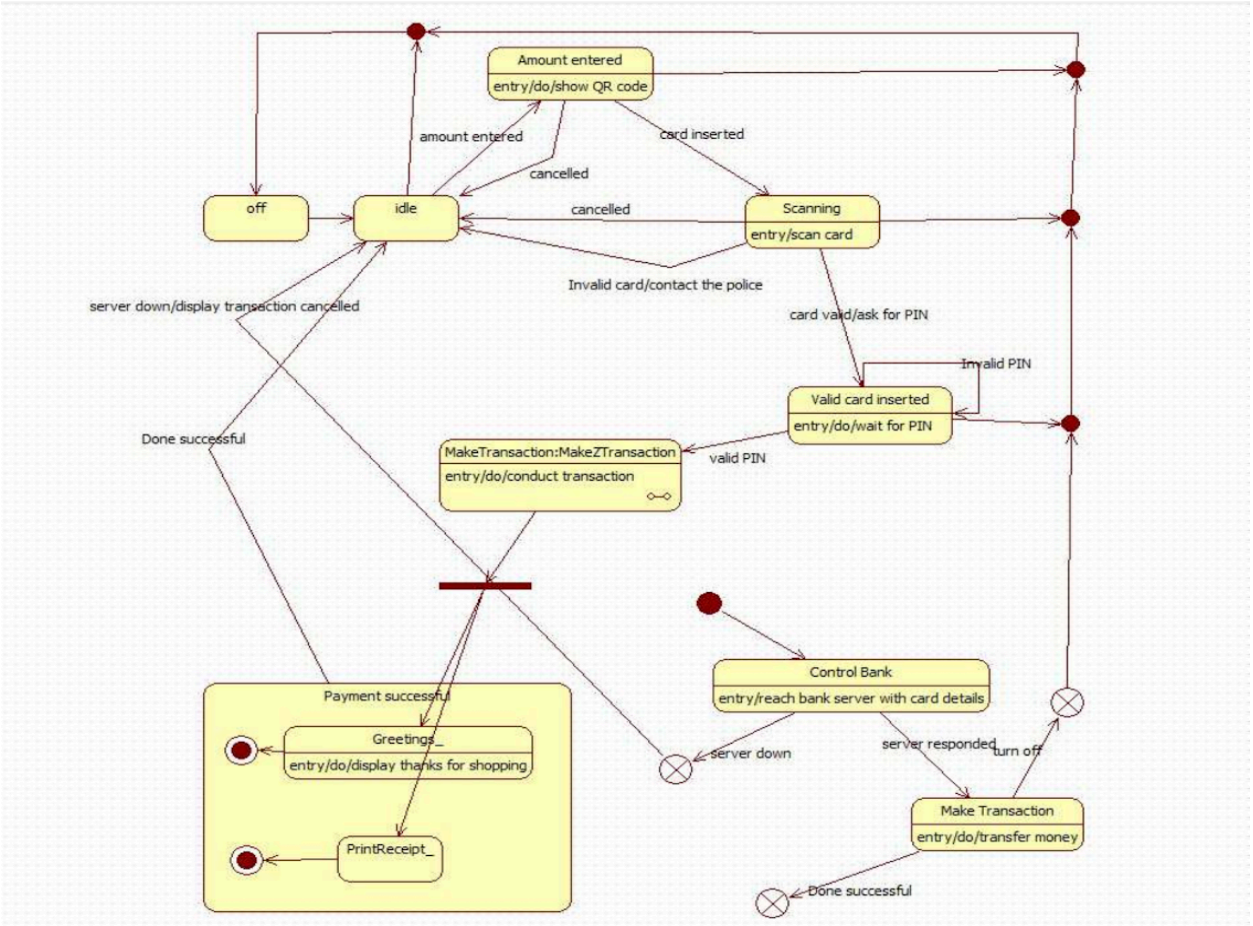
5. Glossary

- PCI-DSS: Payment Card Industry Data Security Standard
- Authorization: Process of approving a payment request
- Capture: Finalizing the transfer of funds
- Refund: Returning funds to cardholder.

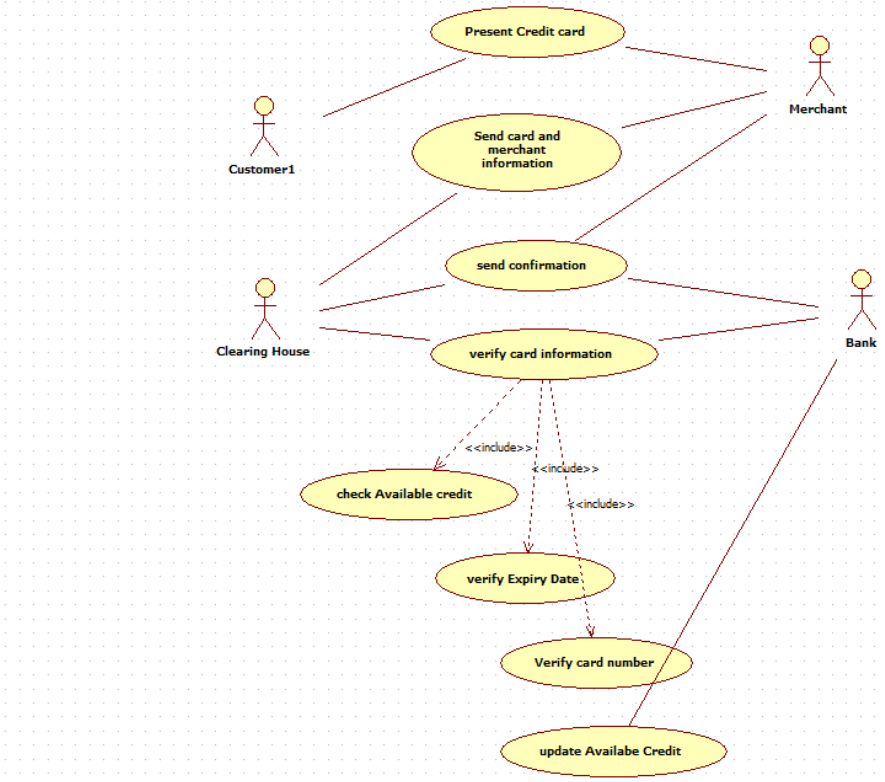
Class Diagram



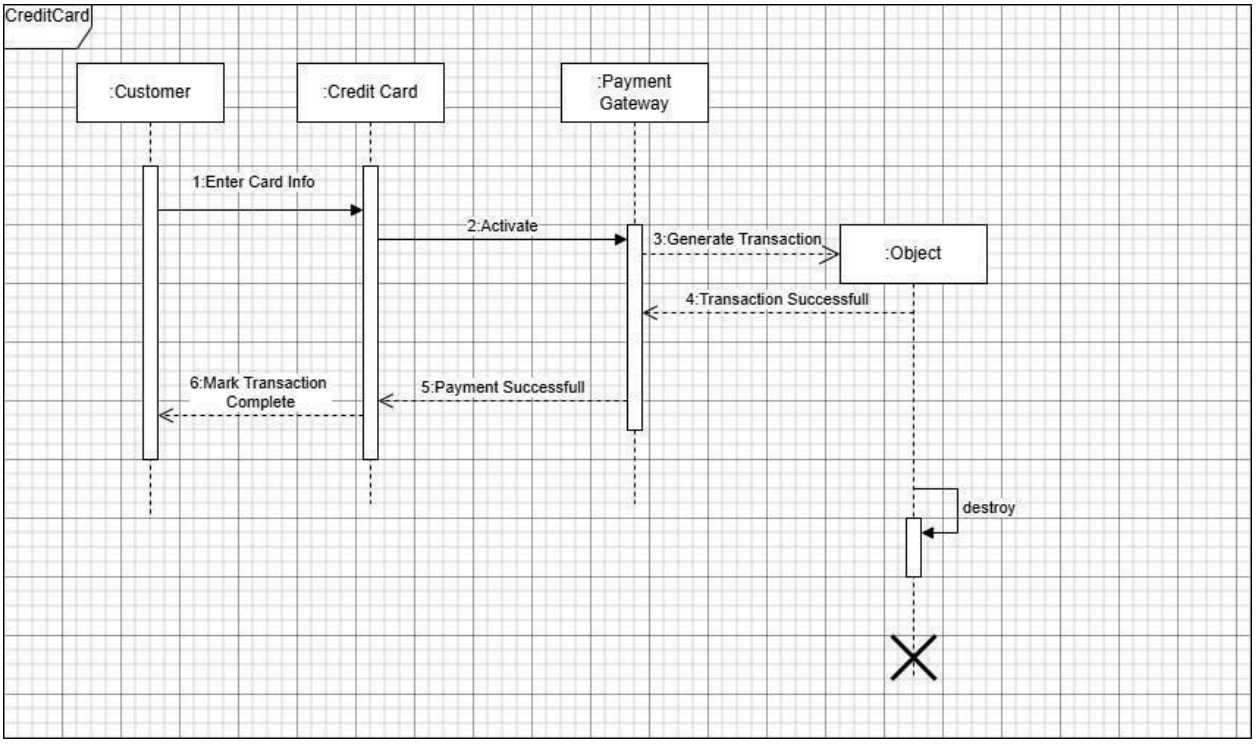
State Diagram



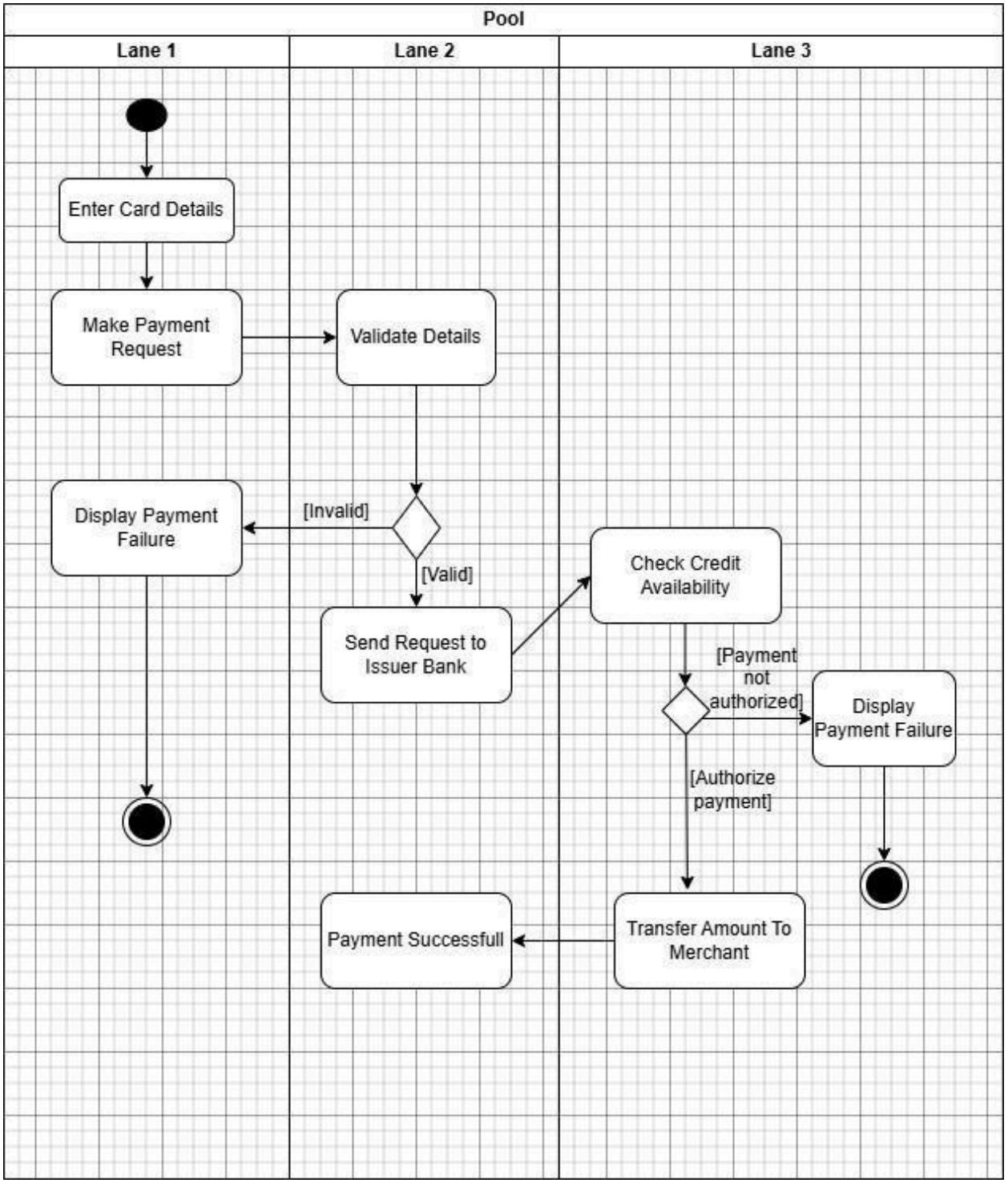
Use Case Diagram



Sequence Diagram

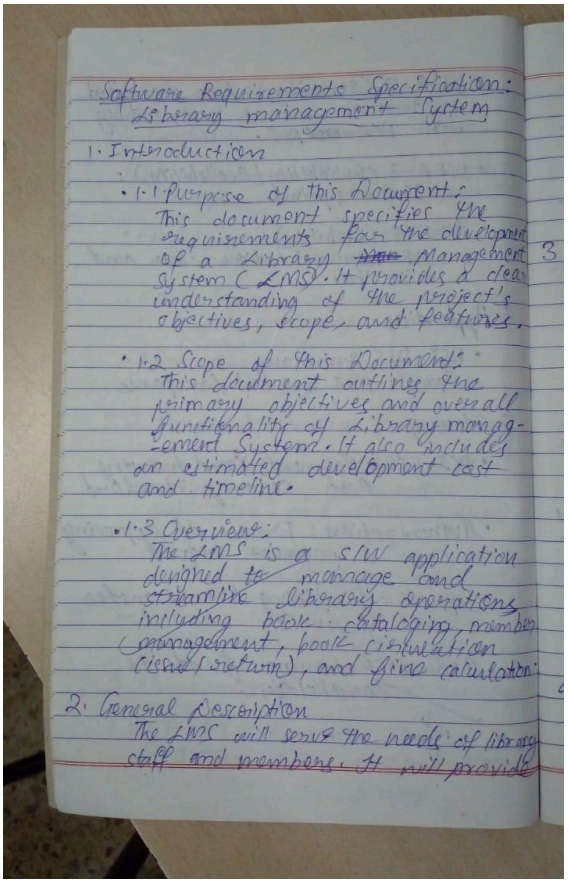


Activity Diagram



Library management system

SRS



Software Requirements Specification:
Library management System

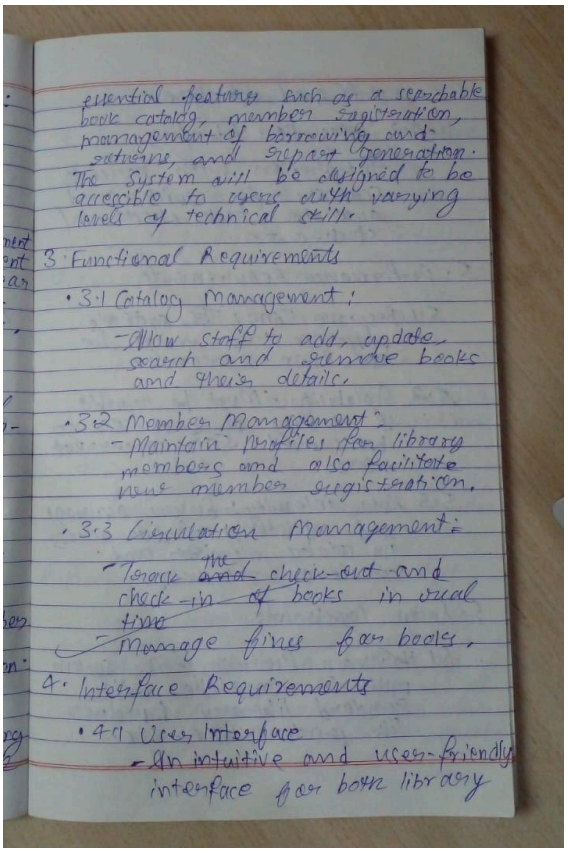
1. Introduction

• 1.1 Purpose of this Document:
This document specifies the requirements for the development of a Library Management System (LMS). It provides a clear understanding of the project's objectives, scope, and features.

• 1.2 Scope of this Document:
This document outlines the primary objectives and overall functionality of Library management System. It also includes an estimated development cost and timeline.

• 1.3 Overview:
The LMS is a s/w application designed to manage and streamline library operations, including book cataloging, member management, book circulation (issue/return), and fine calculation.

2. General description
The LMS will serve the needs of library staff and members. It will provide



essential features such as a searchable book catalog, member registration, management of borrowing and returns, and report generation. The system will be designed to be accessible to users with varying levels of technical skill.

3. Functional Requirements

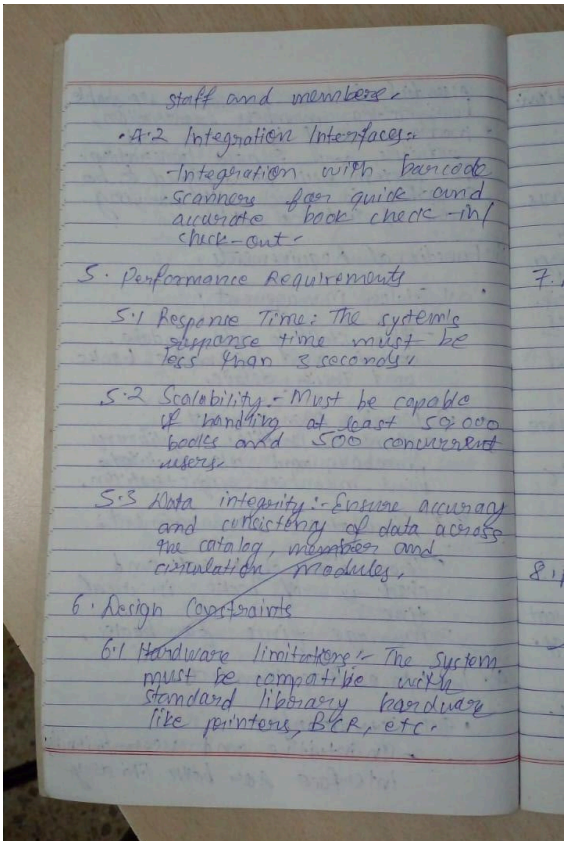
• 3.1 Catalog Management:
- Allow staff to add, update, search and remove books and their details.

• 3.2 Member Management:
- Maintain profiles for library members and also facilitate new member registration.

• 3.3 Circulation Management:
- Track the ~~and~~ check-out and check-in of books in real time
- Manage fines for books.

4. Interface Requirements

• 4.1 User Interface
- An intuitive and user-friendly interface for both library



staff and members.

- 4.2 Integration Interfaces:
  - Integration with barcode scanners for quick and accurate book check-in/check-out.

5. Performance Requirements

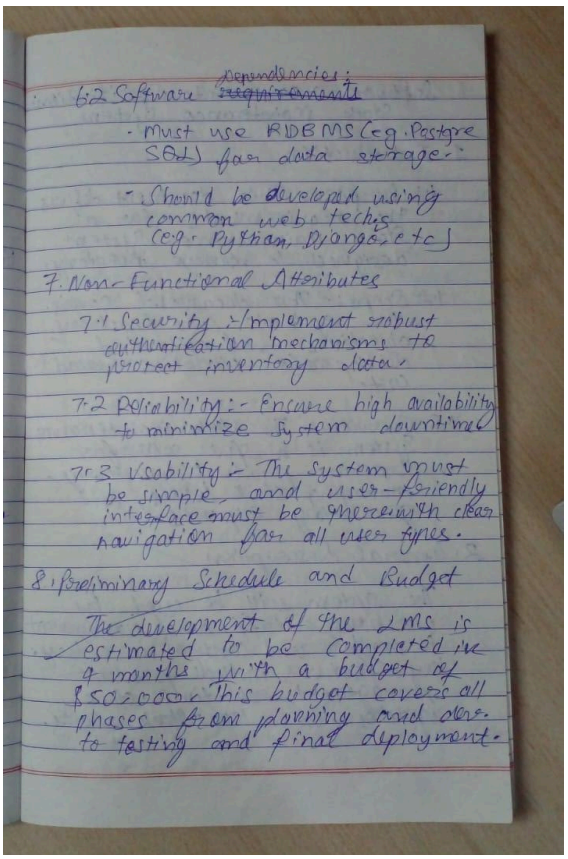
5.1 Response Time: The system's response time must be less than 3 seconds.

5.2 Scalability: Must be capable of handling at least 50,000 books and 500 concurrent users.

5.3 Data integrity: Ensure accuracy and consistency of data across the catalog, members and circulation modules.

6. Design Constraints

6.1 Hardware limitations: The system must be compatible with standard library hardware like printers, BCR, etc.



Dependencies:

6.2 Software requirements

- Must use RDBMS (eg. Postgre SQL) for data storage.
- Should be developed using common web techs (eg. Python, Django, etc.)

7. Non-Functional Attributes

7.1 Security: Implement robust authentication mechanisms to protect inventory data.

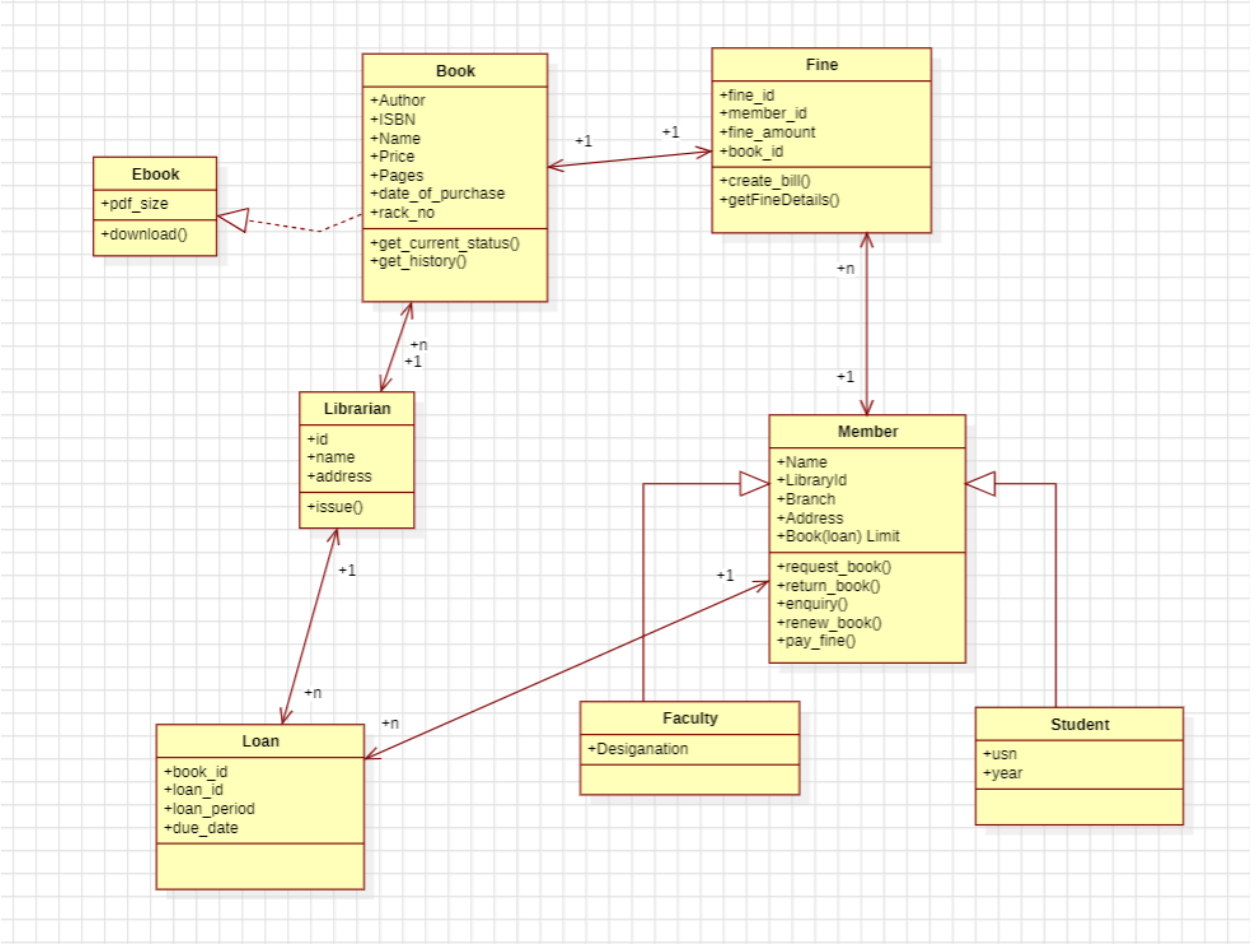
7.2 Reliability: Ensure high availability to minimize system downtime.

7.3 Usability: The system must be simple and user-friendly interface must be there with clear navigation for all user types.

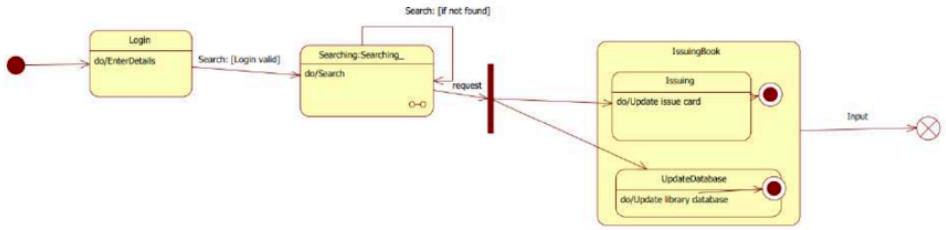
8. Preliminary Schedule and Budget

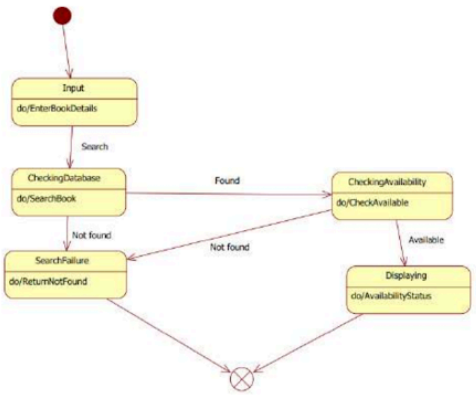
The development of the LMS is estimated to be completed in 4 months with a budget of $50,000. This budget covers all phases from planning and design to testing and final deployment.

Class Diagram

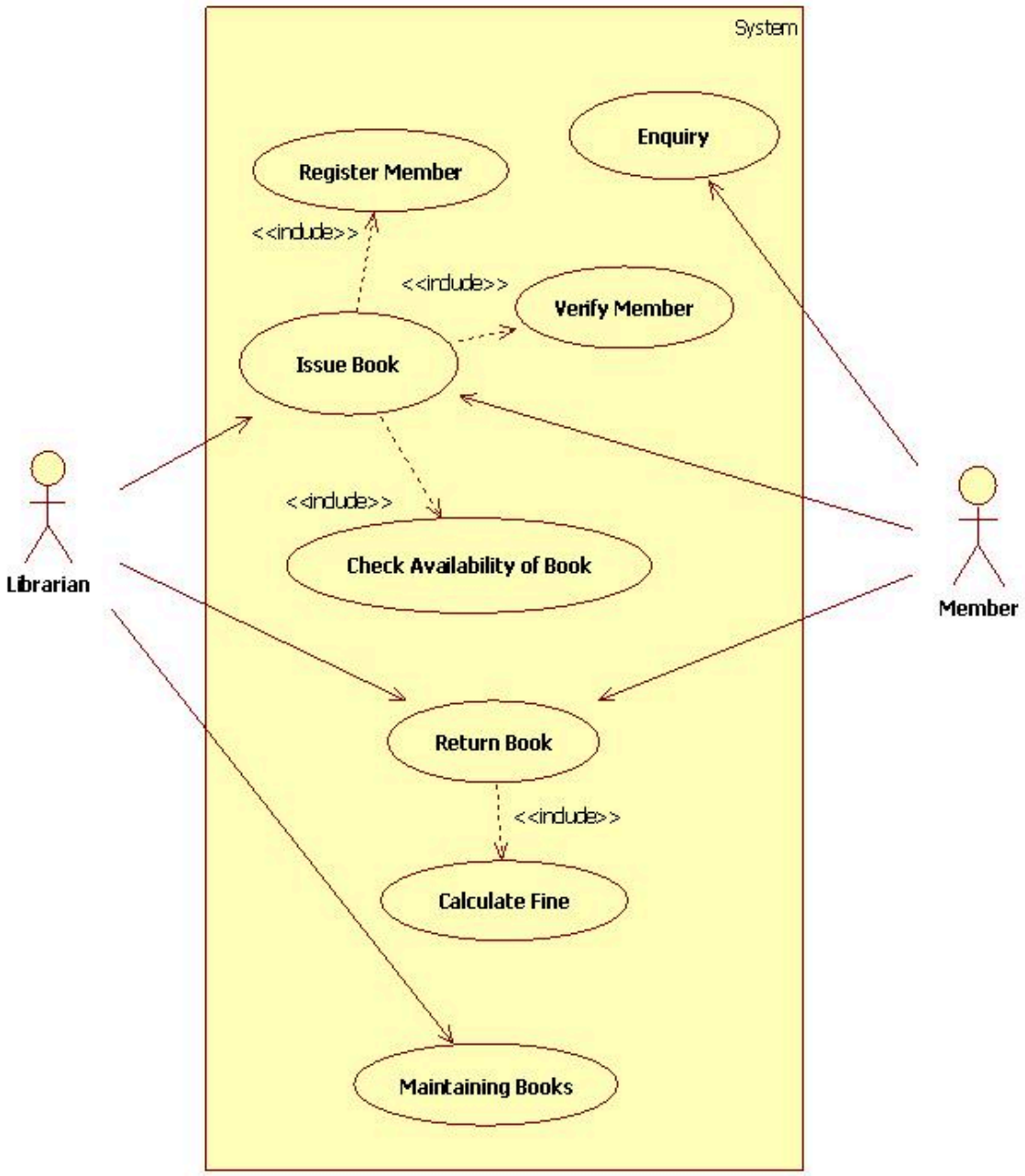


State Diagram

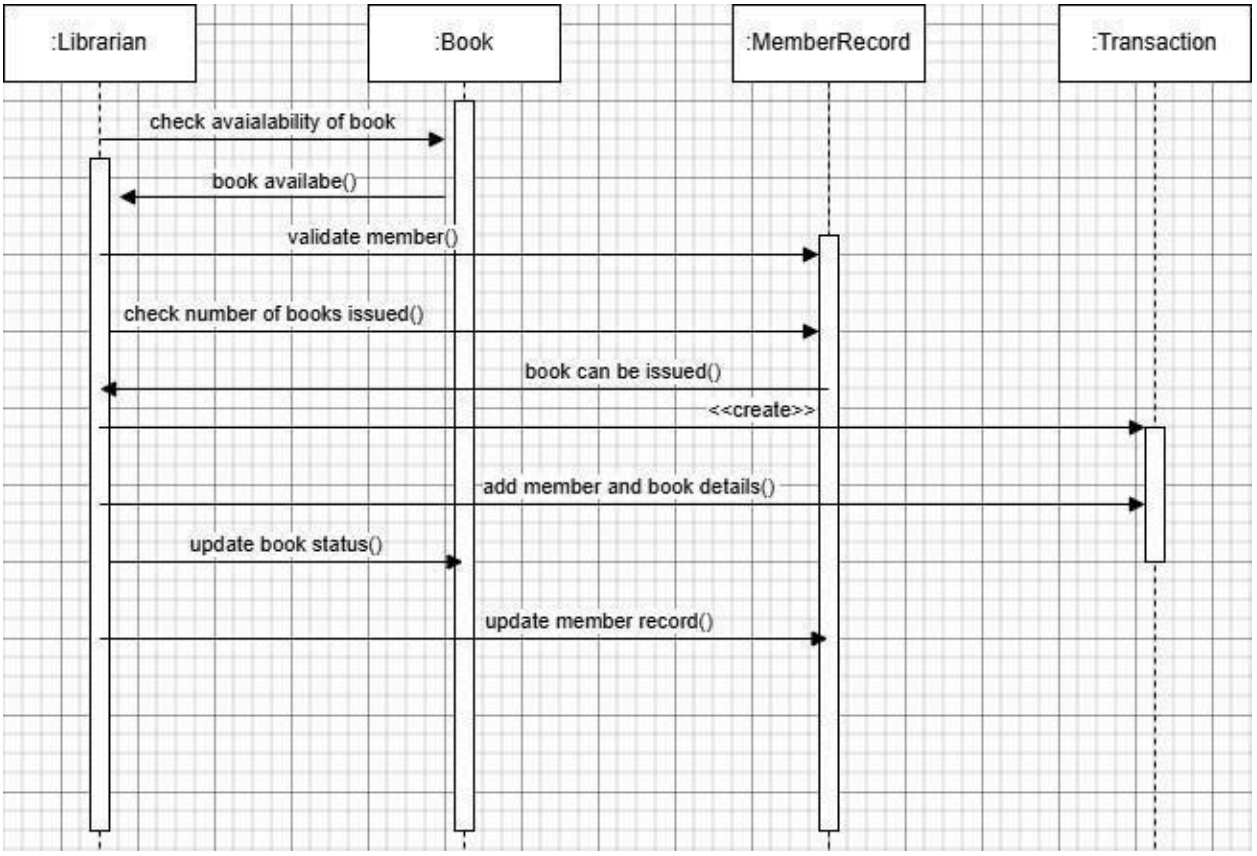




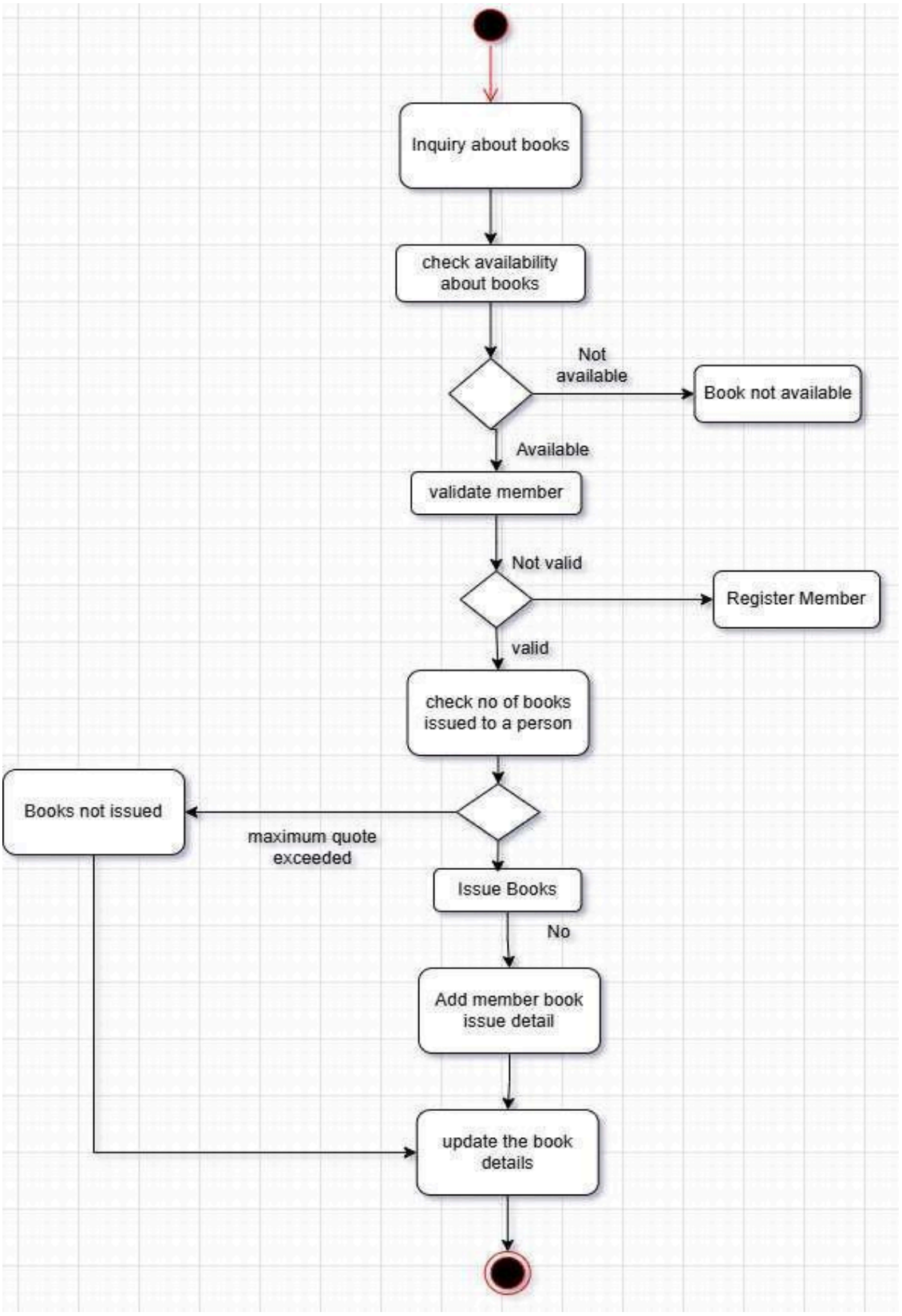
Use Case Diagram



Sequence Diagram

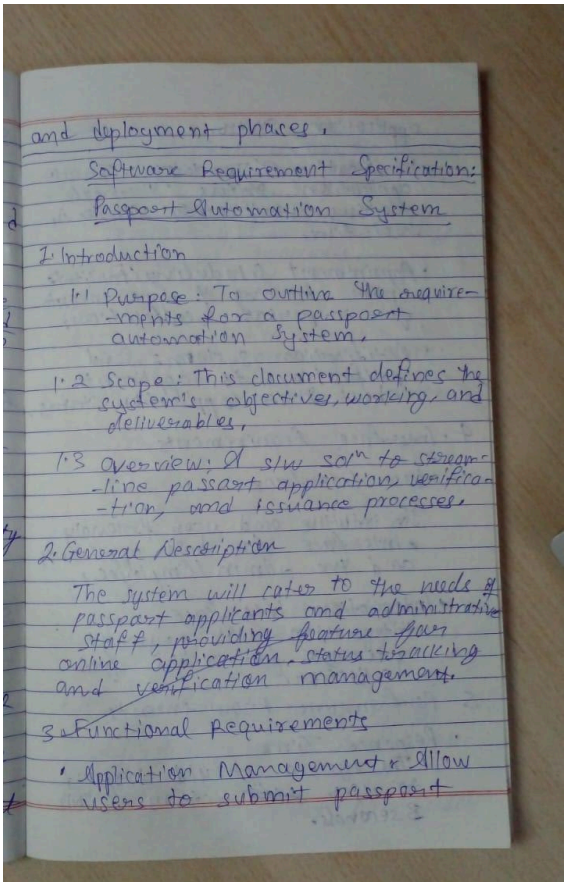


Activity Diagram



Passport automation system

SRS



and deployment phases.

Software Requirement Specification:
Passport Automation System

1. Introduction

1.1 Purpose: To outline the requirements for a passport automation system.

1.2 Scope: This document defines the system's objectives, working, and deliverables.

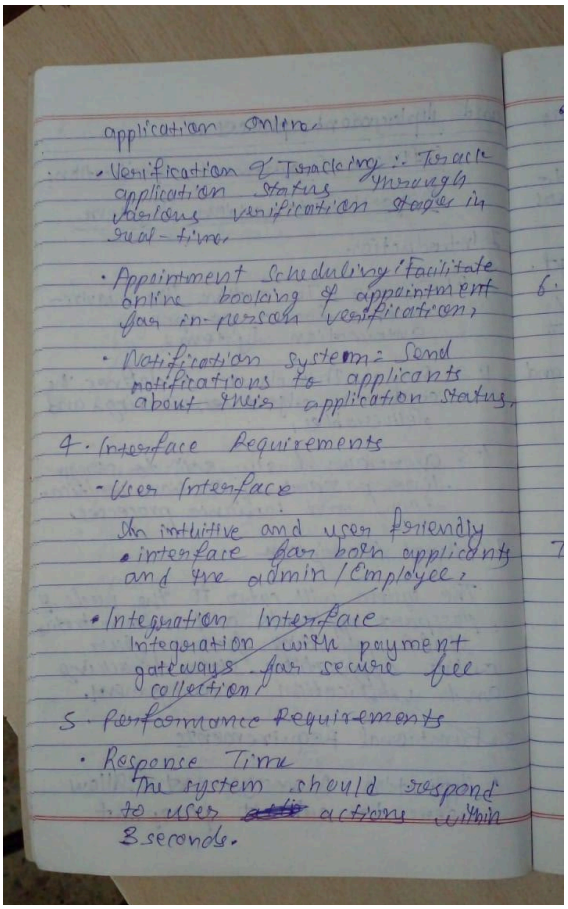
1.3 Overview: A s/w soln to streamline passport application, verification, and issuance processes.

2. General Description

The system will cater to the needs of passport applicants and administrative staff, providing feature like online application, status tracking and verification management.

3. Functional Requirements

• Application Management: Allow users to submit passport



application online.

• Verification & Tracking: Track application status through various verification stages in real-time.

• Appointment Scheduling: Facilitate online booking of appointment for in-person verification.

• Notification System: Send notifications to applicants about their application status.

4. Interface Requirements

• User Interface

An intuitive and user friendly interface for both applicants and the admin/employee.

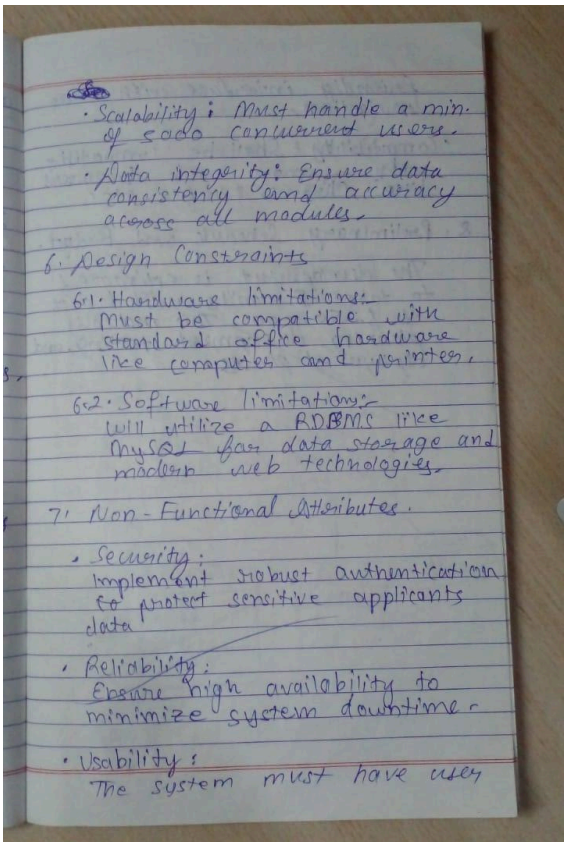
• Integration Interface

Integration with payment gateways for secure fee collection.

5. Performance Requirements

• Response Time

The system should respond to user actions within 3 seconds.



- Scalability: Must handle a min. of 500 concurrent users.
- Data integrity: Ensure data consistency and accuracy across all modules.

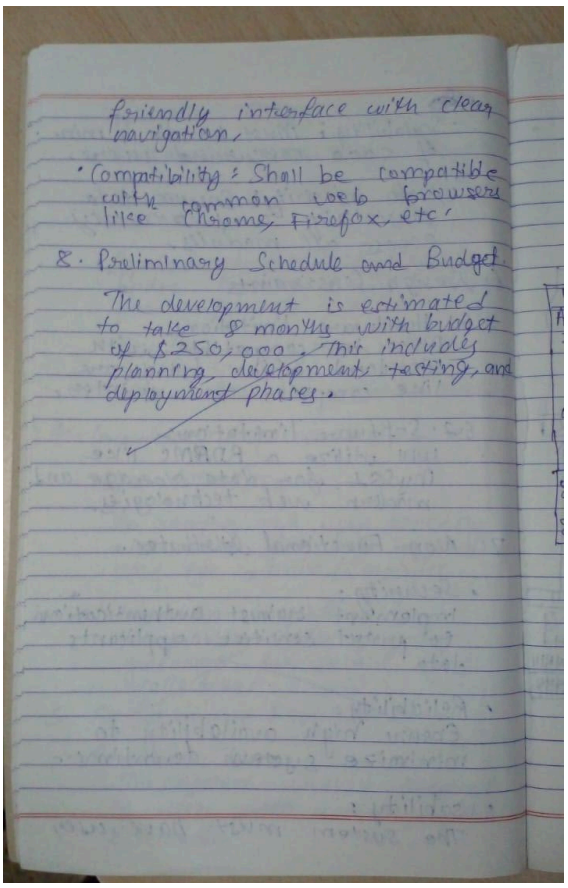
6. Design Constraints

6.1. Hardware limitations:
Must be compatible with standard office hardware like computers and printers.

6.2. Software limitations:
Will utilize a RDBMS like MySQL for data storage and modern web technologies.

7. Non-Functional Attributes.

- Security:
Implement robust authentication to protect sensitive applicants data
- Reliability:
Ensure high availability to minimize system downtime.
- Usability:
The system must have user



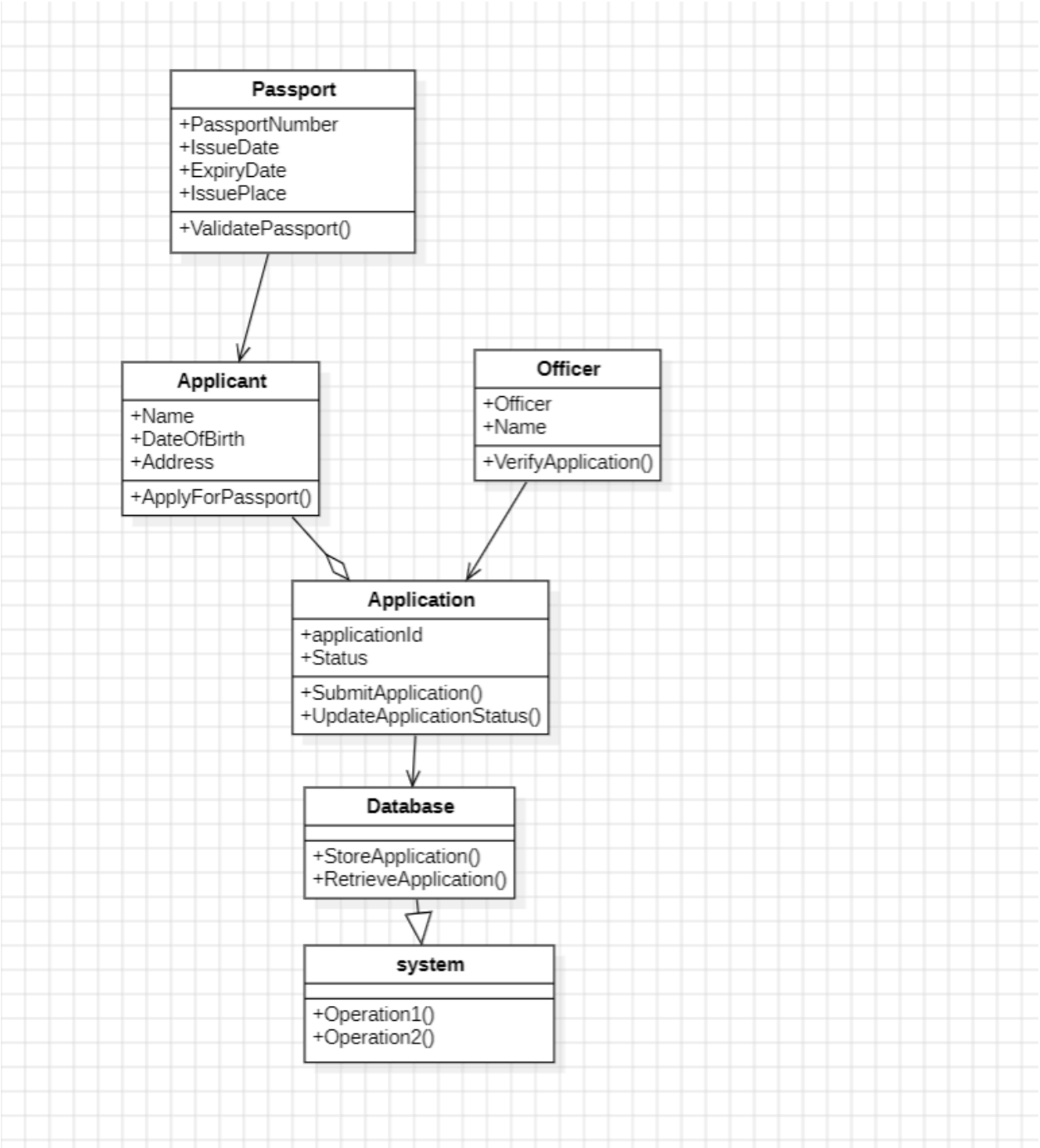
friendly interface with clear navigation.

- Compatibility: Shall be compatible with common web browsers like Chrome, Firefox, etc.

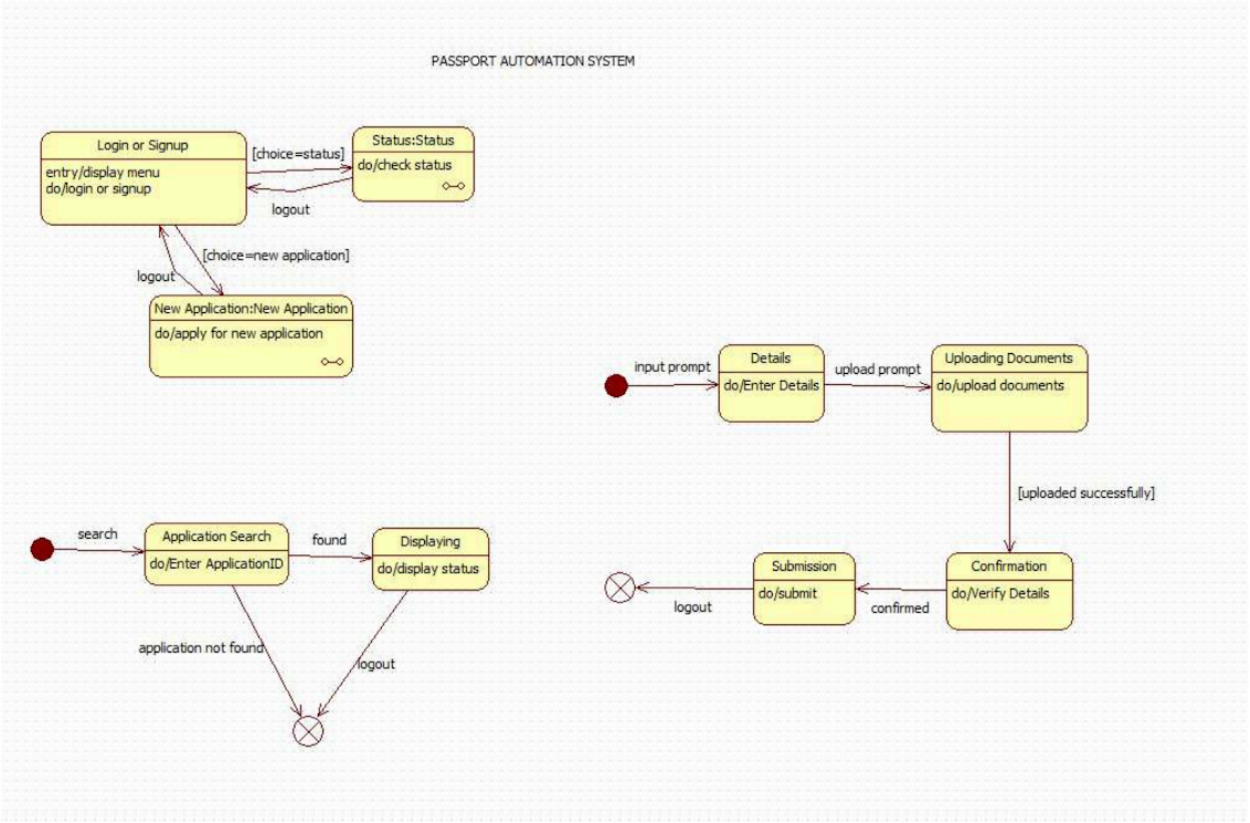
8. Preliminary Schedule and Budget.

The development is estimated to take 8 months with budget of $250,000. This includes planning, development, testing, and deployment phases.

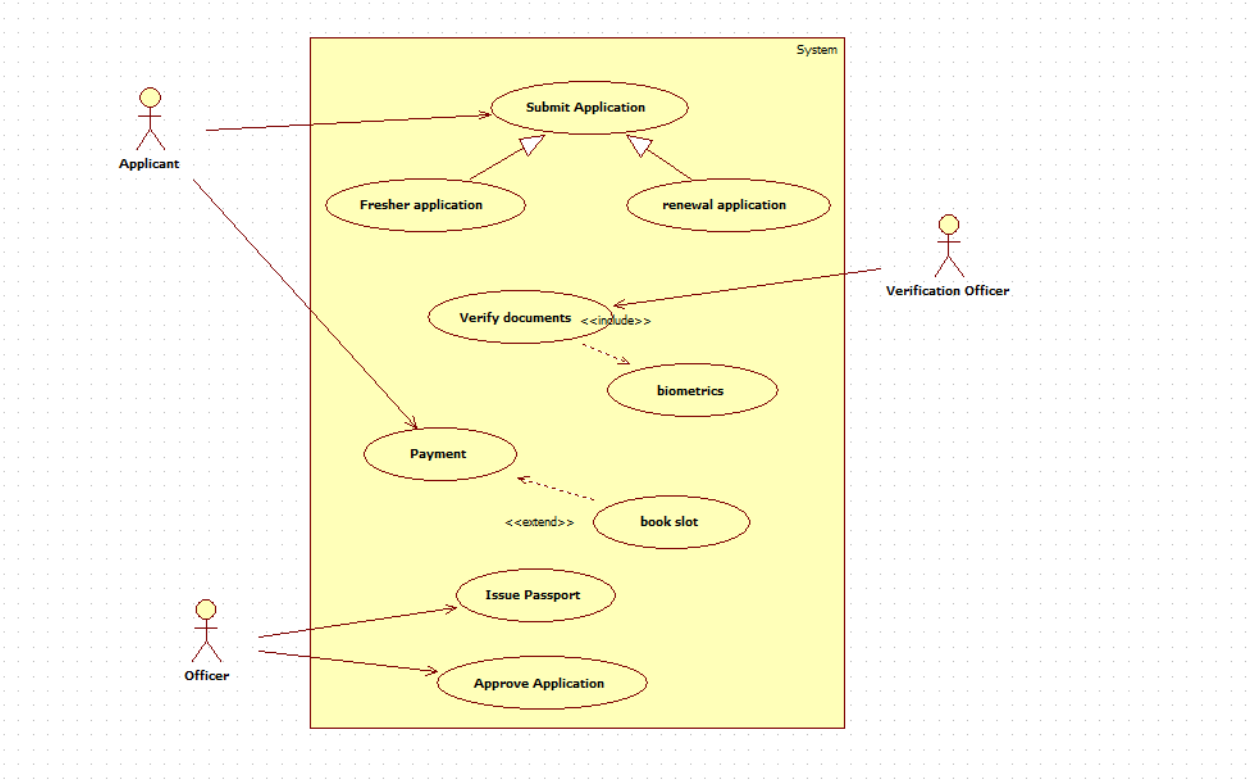
Class Diagram



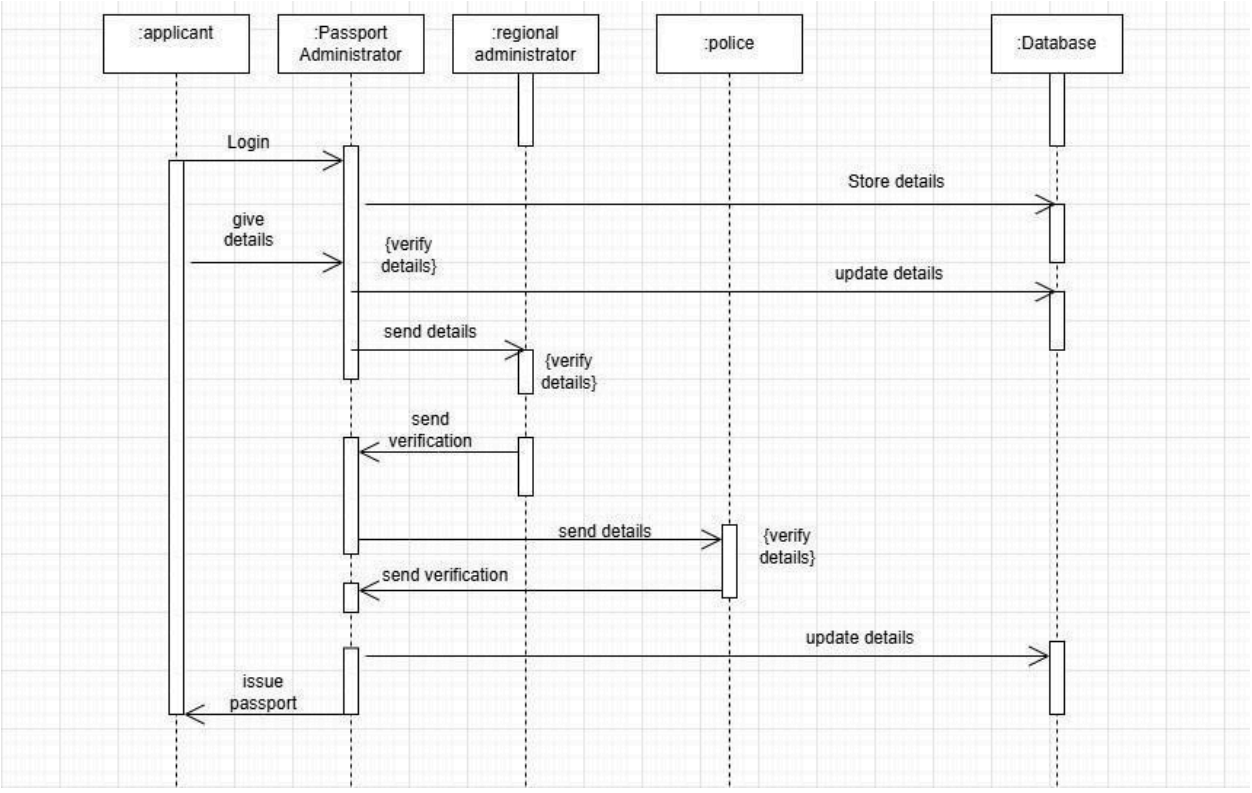
State Diagram



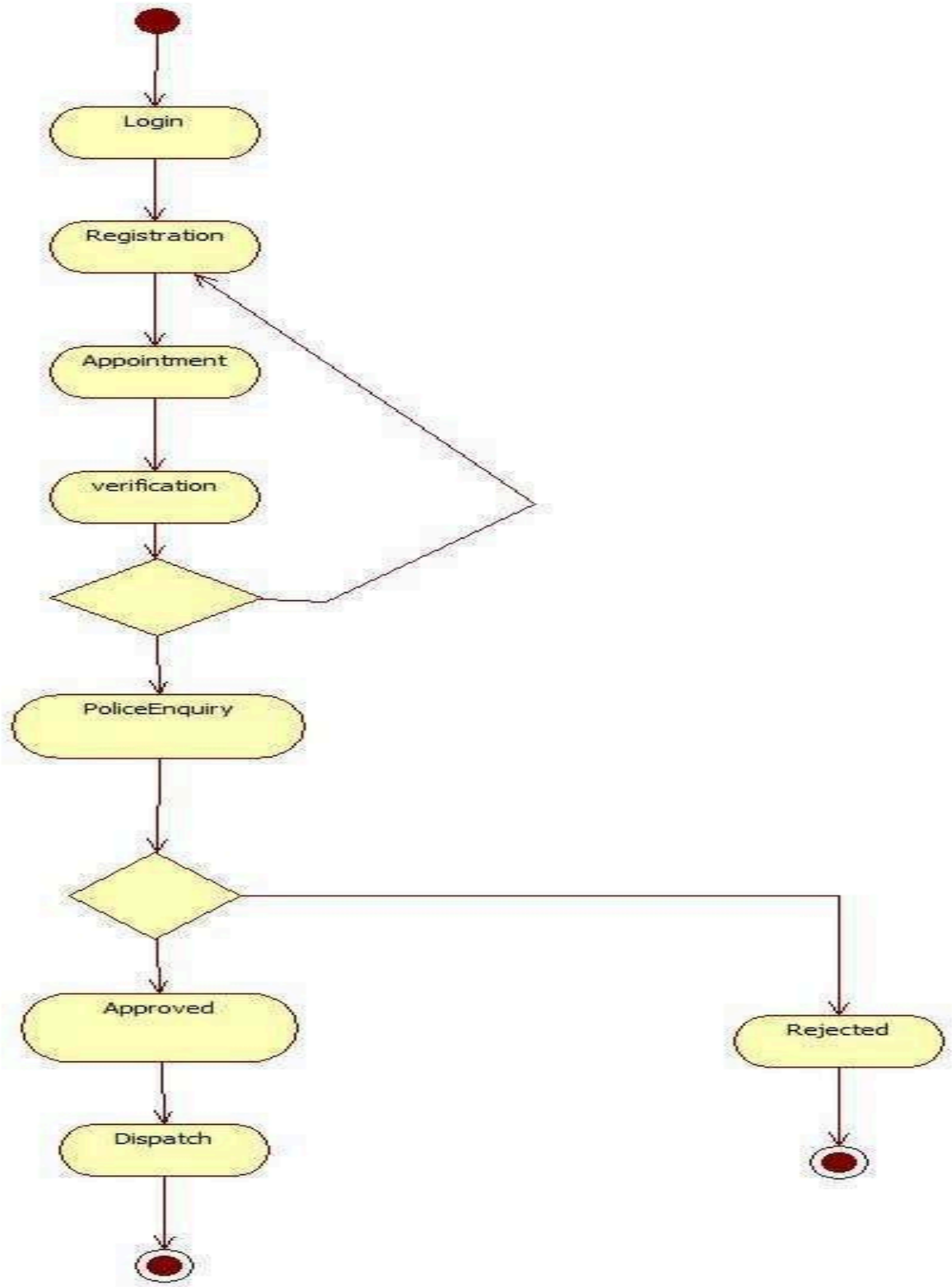
Use Case Diagram



Sequence Diagram

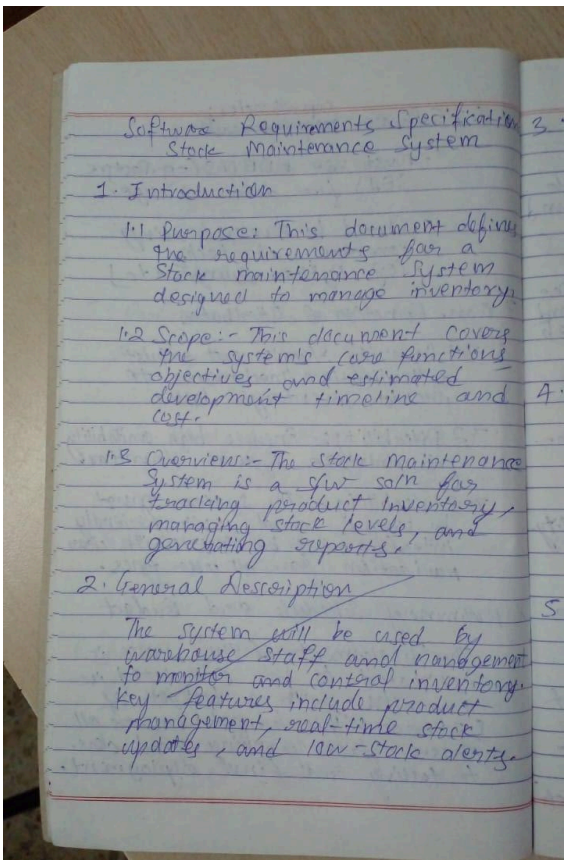


Activity Diagram



Stock maintenance system

SRS



Software Requirements Specification
Stock Maintenance System

1. Introduction

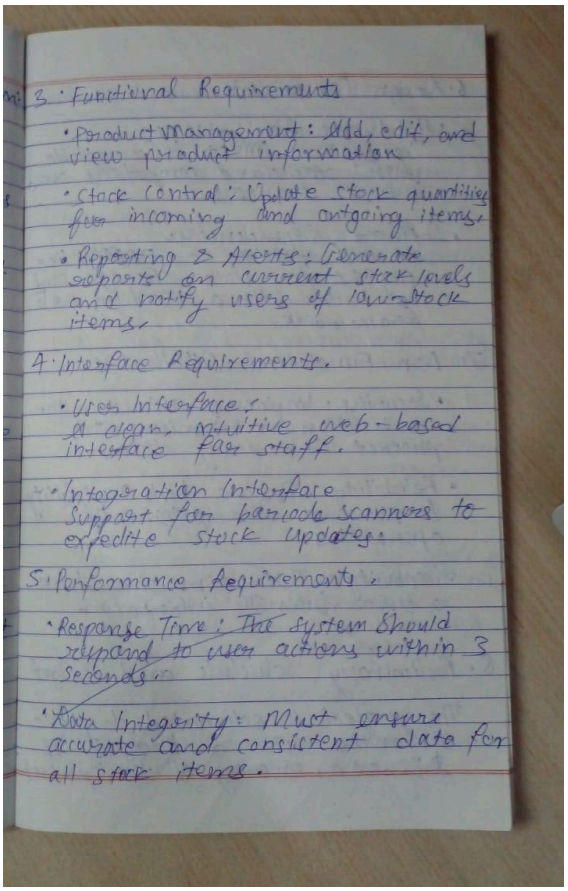
1.1 Purpose: This document defines the requirements for a Stock maintenance System designed to manage inventory.

1.2 Scope:- This document covers the system's core functions objectives and estimated development timeline and cost.

1.3 Overview:- The Stock maintenance System is a SW soln for tracking product inventory, managing stock levels, and generating reports.

2. General Description

The system will be used by warehouse staff and management to monitor and control inventory. key features include product management, real-time stock updates, and low-stock alerts.



3. Functional Requirements

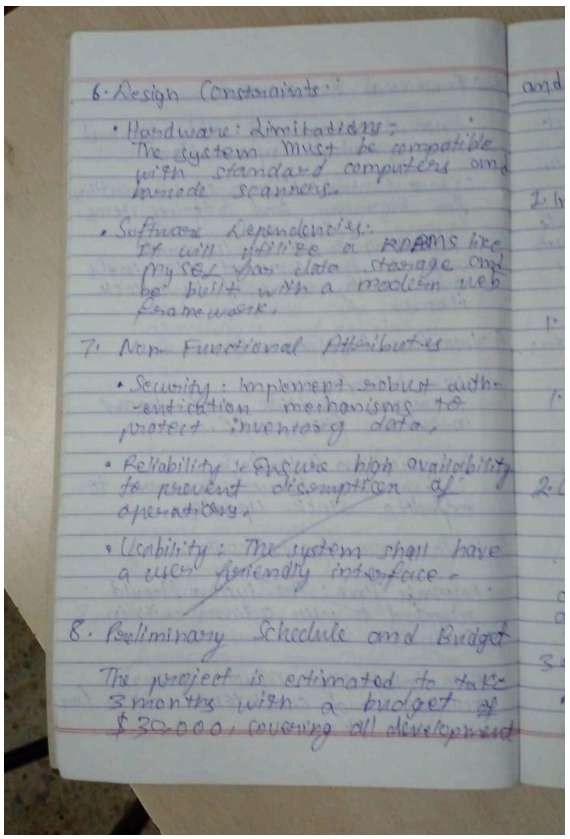
- Product Management: Add, edit, and view product information
- Stock Control: Update stock quantities for incoming and outgoing items.
- Reporting & Alerts: Generate reports on current stock levels and notify users of low-stock items.

4. Interface Requirements.

- User Interface: A clean, intuitive web-based interface for staff.
- Integration Interface Support for barcode scanners to expedite stock updates.

5. Performance Requirements.

- Response Time: The system should respond to user actions within 3 seconds.
- Data Integrity: Must ensure accurate and consistent data for all stock items.



6. Design Constraints:

- Hardware Limitations:
  The system must be compatible with standard computers and barcode scanners.
- Software Dependencies:
  It will utilize a RDBMS like MySQL for data storage and be built with a modern web framework.

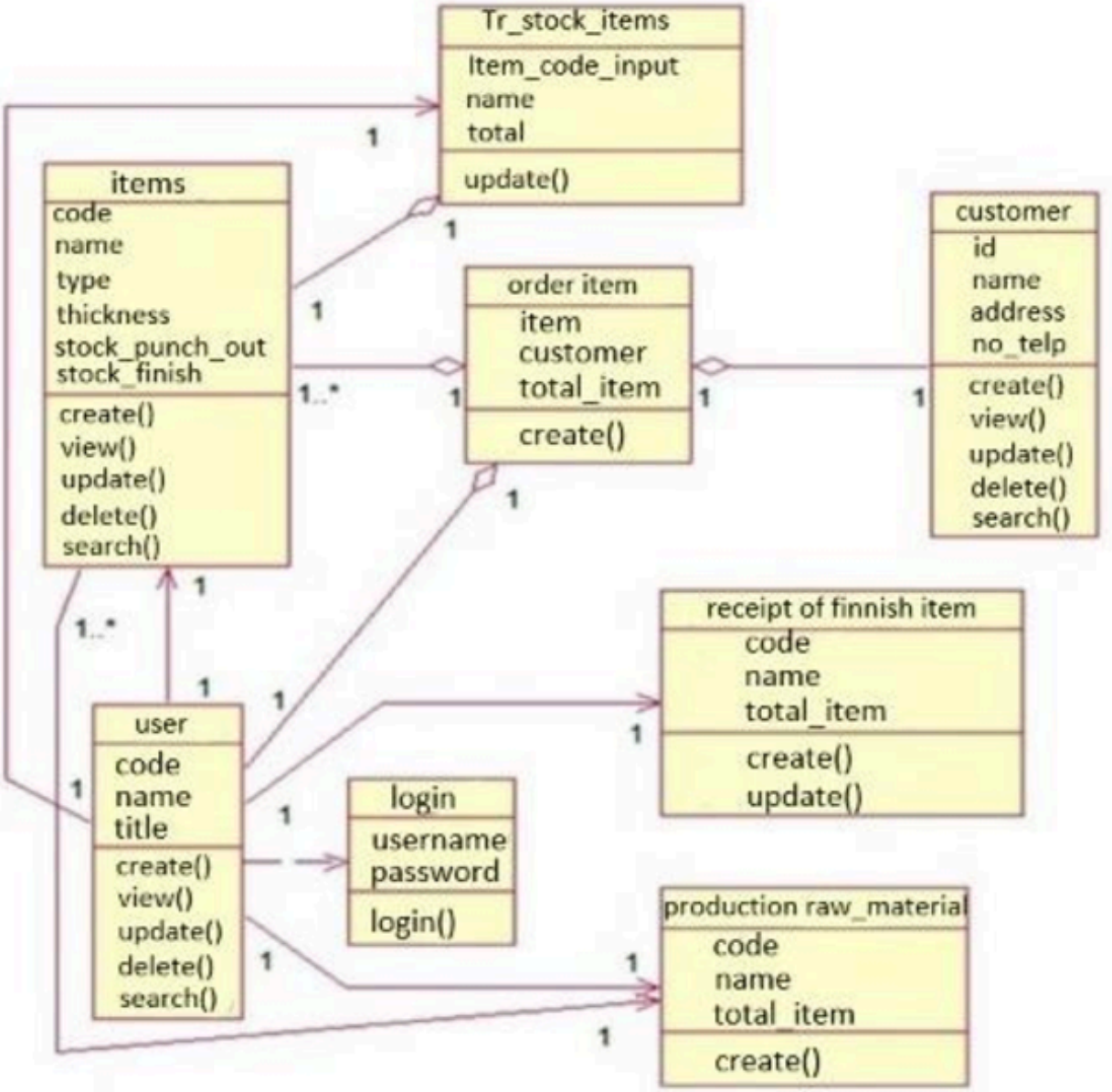
7. Non-Functional Attributes

- Security: Implement robust authentication mechanisms to protect inventory data.
- Reliability: Ensure high availability to prevent disruption of operations.
- Usability: The system shall have a user-friendly interface.

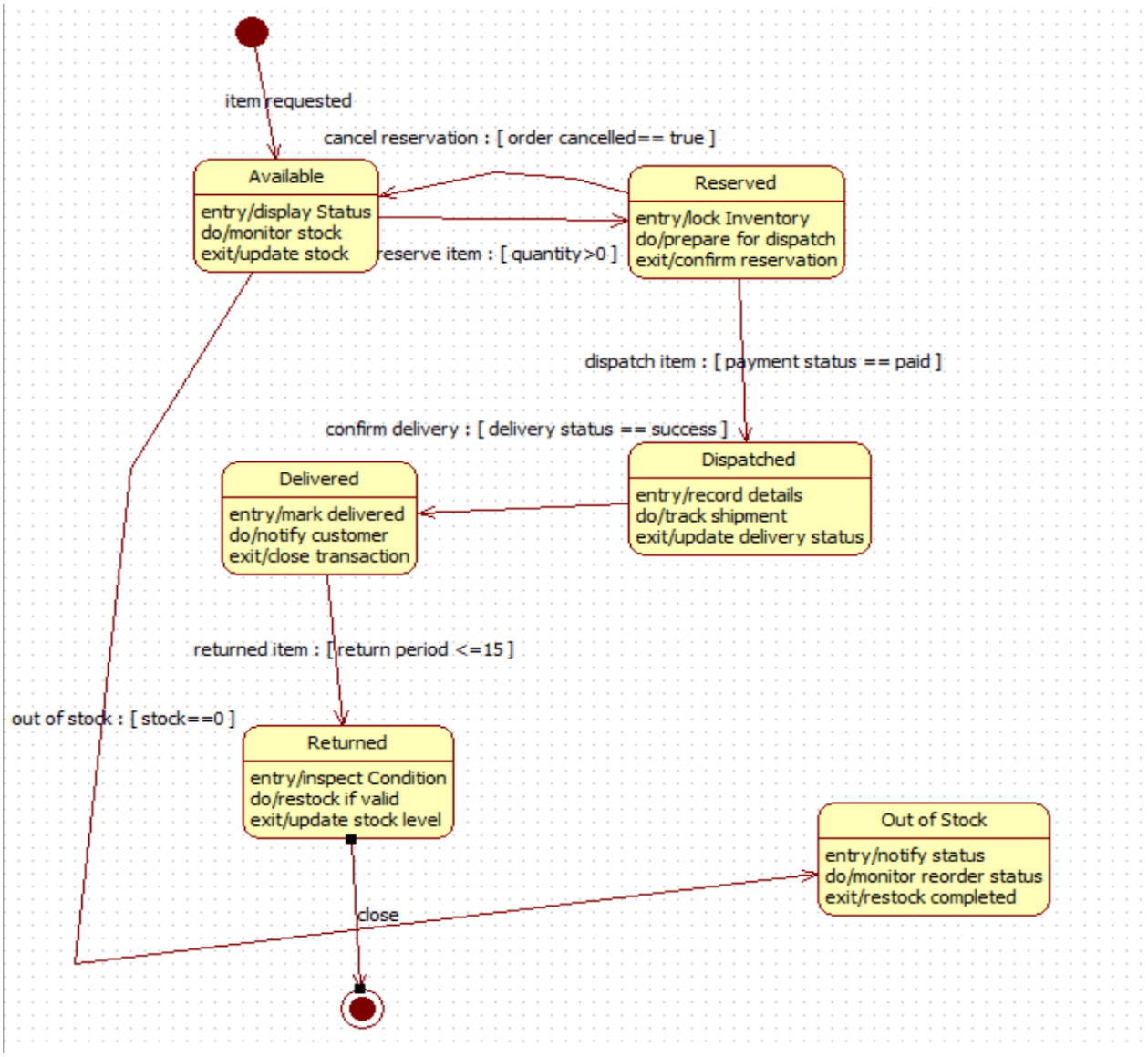
8. Preliminary Schedule and Budget

The project is estimated to take 3 months with a budget of $30,000, covering all development

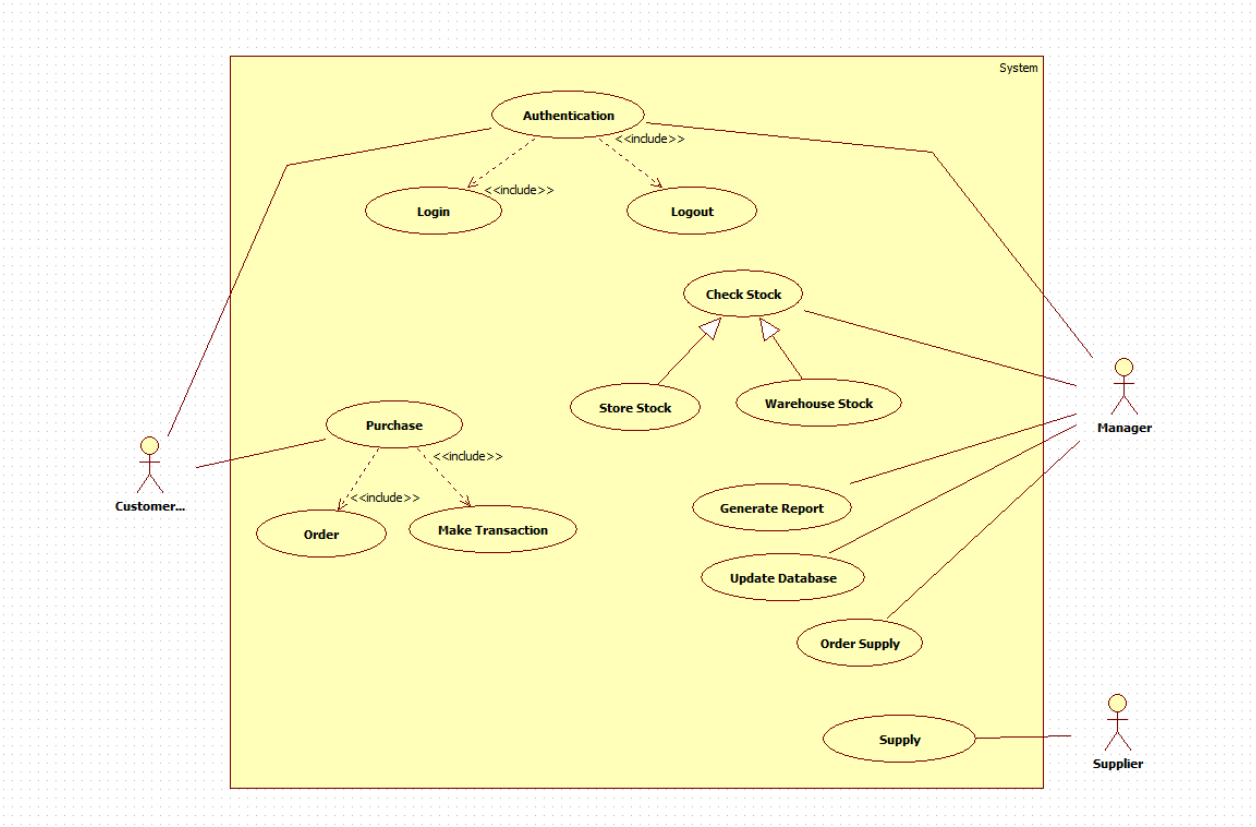
Class Diagram



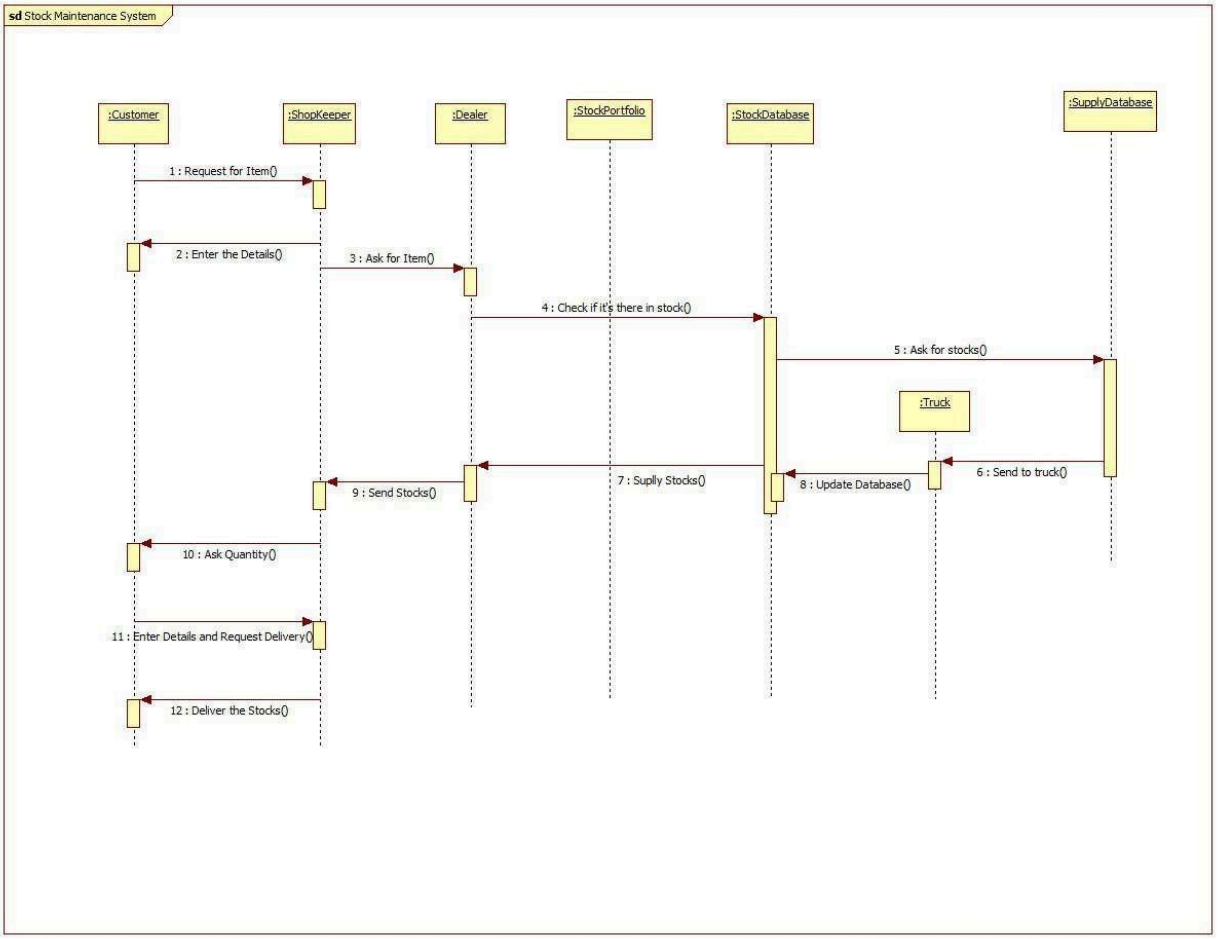
State Diagram



Use Case Diagram



Sequence Diagram



Activity Diagram