Date 30/9/24

LAB-1

Software Requirement Specification

1) Introduction:

1.1) Purpose: To establish a working credit card based online payment system.

1.2) Scope: Authenticate an existing user or create a new user in order to facilitate secure online payment.

1.3) Overview: An automated secure payment gateway system primarily used by the users to pay using their credit cards.

2) General description:

: The credit card based online payment gateway system must enable a user to not only pay using credit cards but also receive certain benefits such as tokens or cashback incentives which will increases user engagement. It will and must provide an all in one user experience to encourage new users to register.

3) Functional Requirements:

1) Credit card Management:

A user must be allowed to maintain and manage all their credit cards. (including addition and deletion)

2) Authentication:

A new user registration must be accepted and authenticated and each returning user must verify themselves before using the system.

3) Transactional concurrency:

The amount of money to be debited must be the same amt of money credited, therefore maintaining transactional concurrency.
No conflict must occur.

4) Interface Requirements:

1) Web Interface:

A website which is compatible on all devices.
The frontend interacts with the user and the backend consists of a database which stores all previous transactions and details of the user.

5) Performance Requirements:

→ Speed: Transactions must be completed quickly and must not take a lot of time.

→ Availability: Huge number of transactions must not cause the website server to crash.

6) Design Constraints:

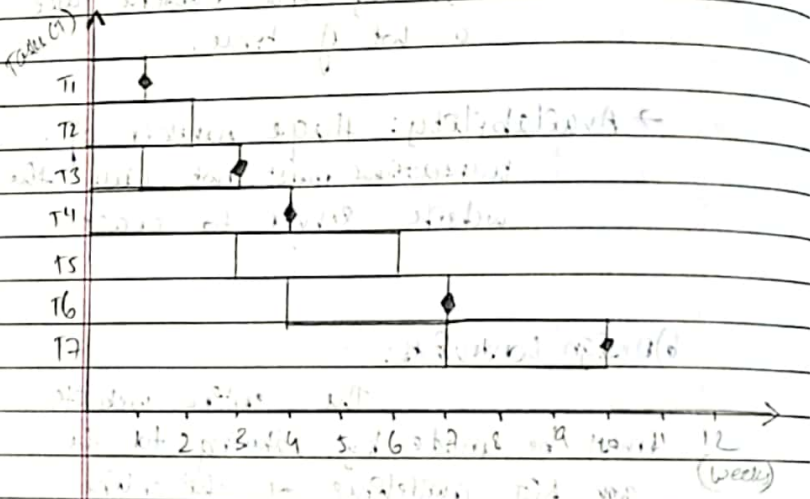
The entire website must be made by adhering to the govt RBI guidelines of the online payments and the software to be used for the same must be implemented using MERN stack.

7) Non-Function Attributes:

1) Security
2) Compatibility

→ Data Integrity
→ Scalable.
→ Improvable.
→ Upgradable.
→ Portable.

8) Preliminary Schedule and Budget:



The entire workload is devided into 7 tasks and each task is to be completed as per the schedule given above.

A total of 5 milestones must be achieved.

In total, the entire project must be completed at the end of 10 weeks.

Budget:

Total Lines of code: 220,000, to be written for the system.

organisation average productivity: $ 120/hr.

Cost per line of code: $\frac{220000}{120} = \frac{5500}{3}$

Total cost: LOC x cost per LOC

$= 220000 \times \frac{5500}{3}$

Total cost = 403333333.3

LAB - 2

7/10/24

Hotel Management System:

1) Introduction:

1.1) Purpose: To establish the functional and non-functional requirements for a Hotel Management System

1.2) Scope: The Hotel management system will automate the hotel booking process, managing customer details, and billing related to the clients.

1.3) Overview: The Hotel management System will automate the entire booking system and management. The system will be quick and responsive for optimal user interaction.

2) General Description:

The automated system for Hotel management is a standalone model which allows the user to login and verify themselves as the first step after which the main functionalities are given to the user. The System features are as follows:

→ Room Booking System: Allows customers to book rooms, orders or via staff entry

→ Check in / check out system: Track guest arrivals and departures.

→ Staff Management: Manage employee data, roles and schedules.

→ Billing System: Automate invoice generation and payment processing.

3). Functional Requirements:

1) User Registration: The system enables a new user to Register themselves and create a profile and on the other hand also enables an existing user to login to the system.

2). check-in / check-out System: A functionality which allows a user to check-in / check-out of from their respective rooms. This will also enable the staff with the same functionality to clean and maintain rooms.

3) Billing and reporting: The overall bill payment and management of different bills must require a different functionality.

4) Interface Requirements:

1) User-System Interface:

The user interface shall provide an intuitive navigation for booking, checking-in/out and managing rooms.

2) Hardware Interface:

The system must consist of barcode scanners for check-in/out.

5) Performance Requirements:

1) Speed: The system must be quick to respond to user inputs.

2) Memory: The system requires 1TB of memory in order to function efficiently.

6) Design Constraints: The software must consist of a frontend developed by using HTML, CSS, Javascript. The backend must consist of technologies like NodeJS or Django.

7) Non-Functional requirements:

→ Portability: The system must be accessable through any devices thus making it portable.

→ Reliability: The system must be reliable in the sense that it must not crash due to huge traffic of users.

8) Preliminary Schedule and Budget:

Phase	Duration (weeks)
Project Initiation	2
Requirements gathering	4
System Design	3
Development	10
Testing	3
Deployment	2

Budget Breakdown:	Estimated cost(₹)
Personnel cost	200,000
Software & Licenses	10,000
Infrastructure	15,000
Security	10,000
Training	7500
Marketing	2500

LIBRARY MANAGEMENT SYSTEM:

1) Introduction:

1.1) Purpose: To establish the functional and non-functional requirements for a library management system.

1.2) Scope: To create an automated system to manage the smooth functioning of all library operations, including cataloging, book lending, and member management.

1.3) Overview: An Independent system that manages all functions related to running a library. The system will allow users to search for books, borrow and return them, and maintain records of borrowing activities.

2) General description: The library system automates key functions like cataloging, borrowing, returning books and managing member records. It serves both staff and members, making

library operations more efficient. Key features include an intuitive book catalog management system that allows librarians to add, update and categorize books efficiently. Users can search for books by title, author, or genre and reserve them if unavailable.

3) Functional Requirements:

•) Book catalog management:
→ Ability to add, update, or remove books from the system.
→ categorization of books by genre, author, publication date etc.

•) User management:
Users can register and manage their profiles, while librarians handle user records, borrowing history and overdue fines.

•) Reservation System:
Users can reserve unavailable books, and the system notifies them when the books are ready for pick-up.

•) Search functionality:
Users can reserve unavailable books by title, author, genre or ISBN to quickly locate what they need.

4) Interface Requirements:
•) User Interface:
The Library Management System will feature a web-based GUI built with modern front-end tech, for ex. HTML5, CSS3, JavaScript, that enables users to perform operations such as browsing, searchings, and borrowing books.

•) Software Interface:
The system will integrate with a relational database management system (RDBMS), utilizing SQL for efficient storage and retrieval of book and user data.

5) Performance Requirement:
•) Response Time:
The system must ensure optimal performance with a response time under 3 seconds for user queries.

•) Scalability:
The Library Management System should scale effectively to support upto 100000 books and 10,000 users, ensuring no significant performance degradation as the database grows.

•) Error Rate:
The system must maintain high reliability, with an error rate of less than 0.1% for critical operations such as book inventory management, ensuring data integrity and accurate transactions.

6) Design Constraints:
•) Database Requirements:
The system will integrate a relational database (RDMS) like MySQL or PostgreSQL for robust storage and management.

•) Platform Requirements:
The Library System will be developed as a web-application, ensuring compatibility with all major browser (including chrome, Safari etc.).

7) Non-Functional Requirements:
•) Security:
The system should ensure that user data, including passwords is stored in an encrypted format. Only authorised user (Librarians) can modify book or user records.

•) Portability:
The system should be compatible across different operating systems, including Windows, macOS and Linux.

•) Reliability:
The system should have an uptime of 99.9%.

8) Preliminary Schedule and Budget:
•) Schedule:
•) Initial Project Inspection: Initial phase where the project scope and goal are defined.
•) Requirements gathering: Collection of functional and non-functional requirements from stakeholders.

•) System Design:

Structuring the system Architecture and detailed design for implementation.

•) Development:

Actual coding & building of the system based on design specifications.

Schedule:	Duration (weeks):
Project Inspection	2
Requirements Gathering	4
System Design	3
Development	10
Testing	3
Deployment	2

Budget:	cost ($) in dollar:
category	
Personnel cost	200,000
Software Licenses	10,000
Infrastructure	15,000
Security	10,000
Training	7,500
Marketing	2500.

STOCK MAINTENANCE SYSTEM:

1) Introduction:

1.1) Purpose:

The System aims to automate the management of stock inventory for businesses.

1.2) Scope:

The System will allow businesses to manage stock records, monitor level of inventory, automate the reordering process and generate reports on stock transactions.

1.3) Overview:

The System is designed to automate stock tracking and inventory management for businesses.

2) General Description:

The system will provide businesses with real-time stock information and automate key stock management process such as reorder alerts, stock level tracking and report generation.

3) Functional Requirements:

•) User Management:

The system will allow administrators to create, update and manage user accounts with different access levels.

•) Stock Management:

The system will allow users to Add, update, delete stock items and also view current stock levels.

•) Reorder Alerts:

The system will automatically generate and send notifs when stocks levels fall below the reorder threshold.

•) Reporting:

The system will generate reports on stock movements, including stock received, issued and current stock levels.

4) Interface Requirements:

•) User Interface:

The system will provide a web based GUI for users to manage stock, view reports and set reorder threshold.

•) Software Interface:

The system will Integrate with a Relational database (eg. MySQL) to store stock & user data.

5) Performance Requirements:

•) Response Time:

The System should respond to user queries, such as stock searches, within 2 seconds under normal conditions.

•) Scalability:

The system should be able to handle upto 50000 stock items and 5000 users without performance degradation.

•) Error Rate:

The system should maintain an error rate of less than 0.05% for stock Inventory operations.

6) Design Constraints:

•) Database Selection:

The system will use a relational database to store all staff and user data.

•) Platform Requirements:
The system will be a web application compatible with major browsers, including chrome, Firefox etc.

3) Non-Functional Requirements:

•) Security:
The system will store credentials and sensitive data in encrypted format (eg. AES-256).

•) Portability:
The system will be compatible across different operating system, including windows, macOS and Linux.

•) Usability:
The system should have an uptime of 99.9%

8) Preliminary Schedule & Budget:

Schedule:

Phase	Duration (weeks)
Project Inception	2
Requirements Gathering	4
System Design	3
Development	10
Testing	3
Deployment	2

Budget:

Category	Estimated cost ($)
Personell costs	150,000
Software licenses	10,000
Infrastructure	12,000
Security	8,000
Training	5,000

PASSPORT AUTOMATION SYSTEM:

1) Introduction:

1.1) Purpose:

The purpose of this document is to describe the Software Requirements specification (SRS) for the system.

1.2) Scope:

The system is designed to handle the entire lifecycle of passport processing, from online application submission to passport issuance.

1.3) Overview:

This document outlines the requirements for the passport Automation System. It describes the system's functionality, performance requirements etc..

2) General Description:

The system is a web-based platform that simplifies the passport application process. It allows users to apply for passports, submit documents, schedule appointments and so much more.

3) Functional Requirements:

•) User Registration and Login:

Applicants can create accounts, log in, and reset passwords if needed.

•) Passport Application:

Users can fill out the application form, upload required docs, and submit them online.

•) Appointment Scheduling:

Applicants can schedule appointments for document verification or biometric data collection.

•) Application Status Tracking:

Users can track their application status in real-time.

4) Interface Requirements:

•) User Interface:

A user-friendly interface for applicants to submit applications, track status and schedule appointments.

•) Admin Interface:

Integration with external govt. databases for identity verification.

5) Performance Requirements:

•) Response Time:

The system should respond to user requests within 3 seconds.

•) Error Rate:

The system should maintain an error rate of less than 0.1%.

6) Design Constraints:

•) Database Selection:

A relational database will be used to store applicant information, documents uploads etc..

•) Platform Requirements:

The system will be developed as a web application, compatible across multiple browsers.

7) Non-functional Requirements:

•) Security:

User data, including personal information, must be securely encrypted.

•) Usability:

The system should offer a simple, intuitive user interface for both applicants and administrators.

8) Preliminary Schedule and Budget:

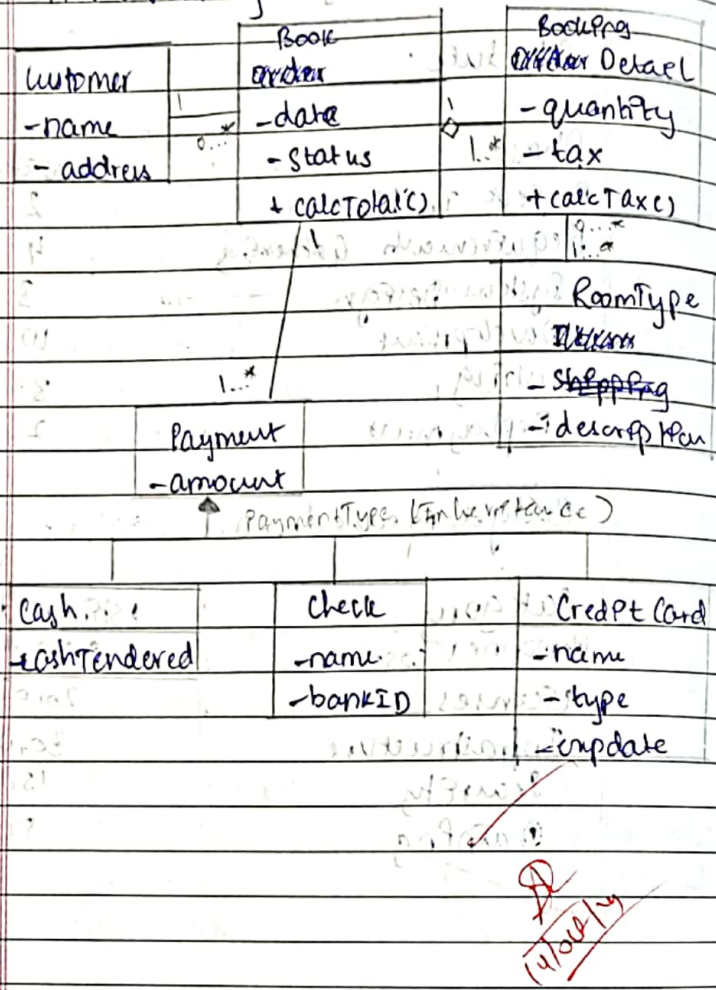
Phase	Duration (weeks)
Project Initiation	2
Requirement Gathering	4
System Design	3
Development	10
Testing	3
Deployment	2

Budget:

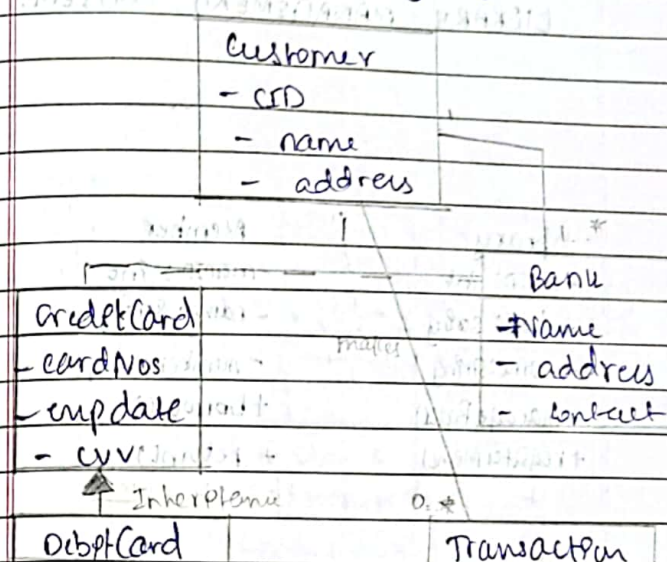
Category	Cost ($)
Personnel cost	200,000
Software licenses	10,000
Infrastructure	15,000
Security	10,000
Training	7,500
Marketing	2,500

Class Diagram:

Hotel management:

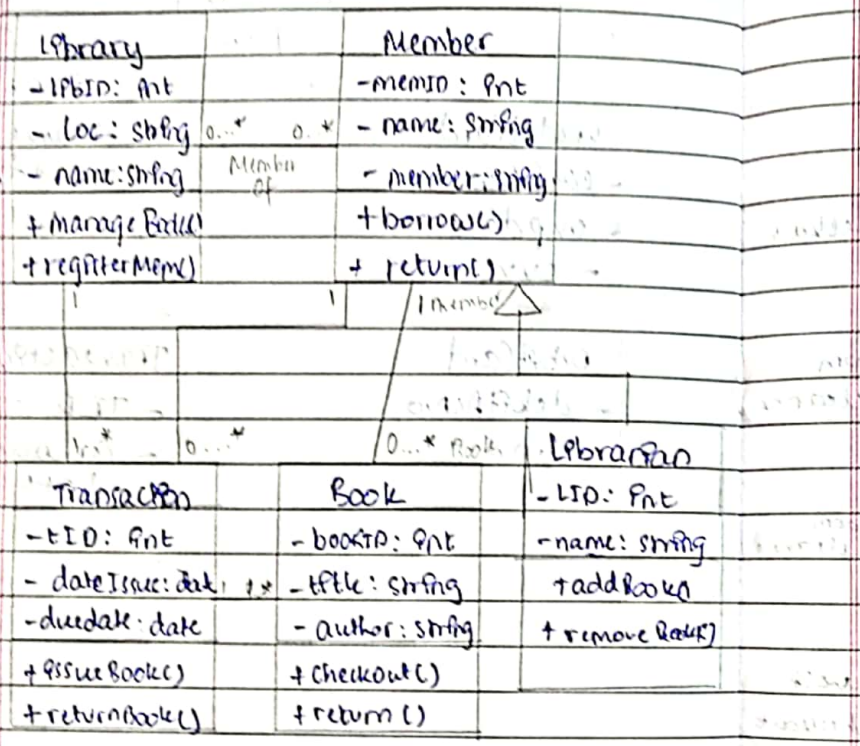


Credit card processing System:

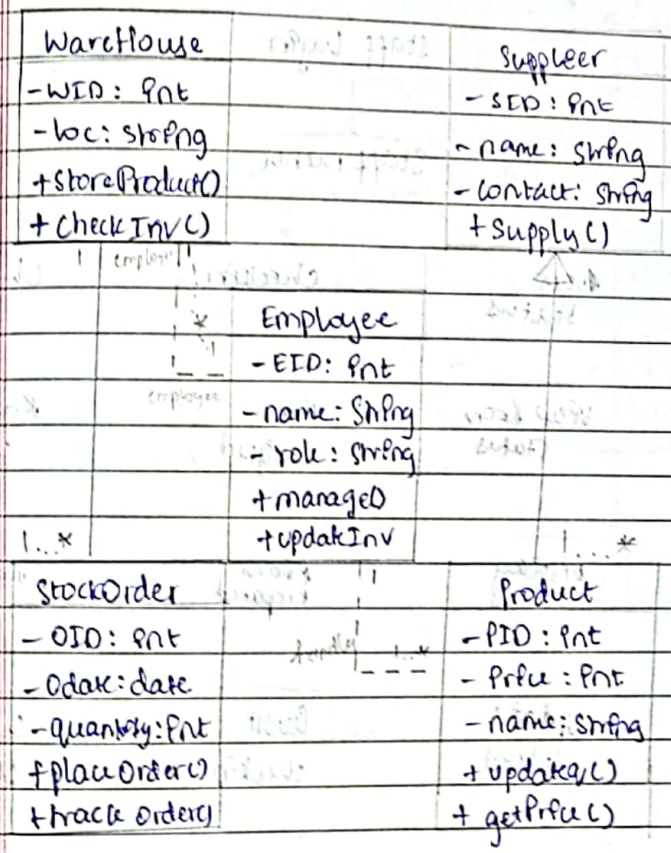


LAB - 4

LIBRARY MANAGEMENT SYSTEM.



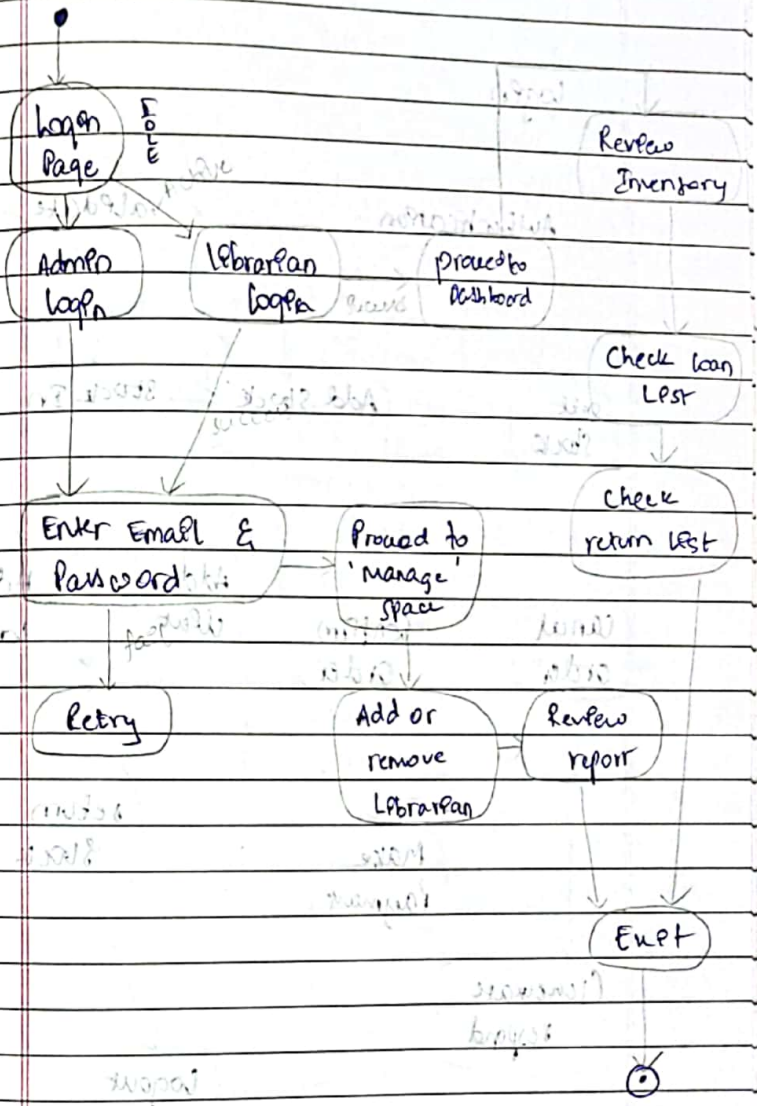
STOCK MANAGEMENT SYSTEM.



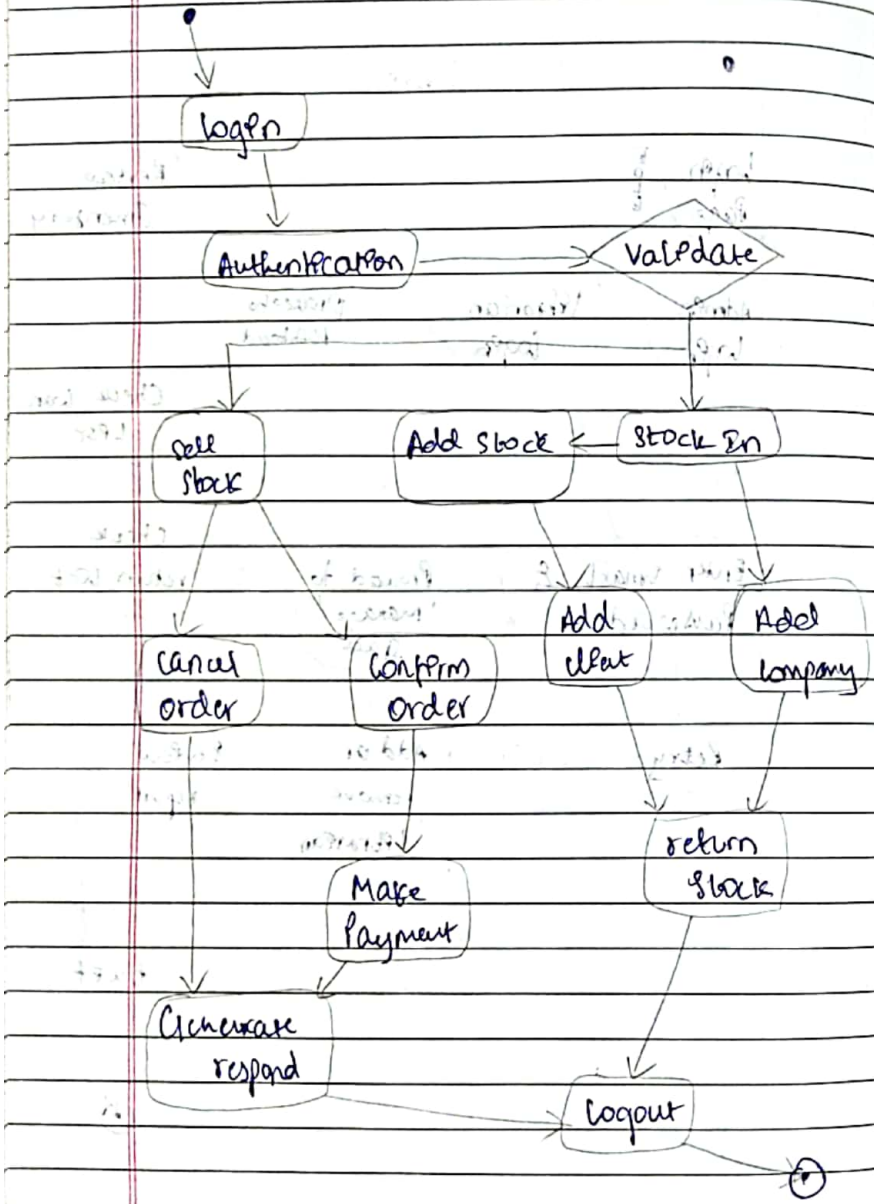
LAB STATE DIAGRAMS

Hotel Management System.

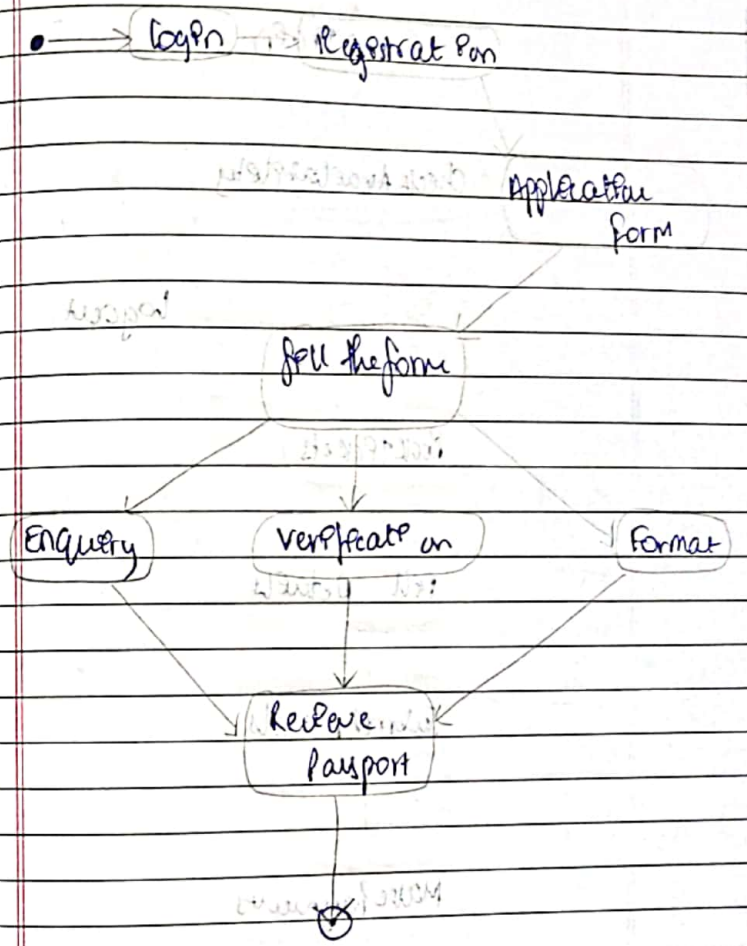
LIBRARY Management SYSTEM



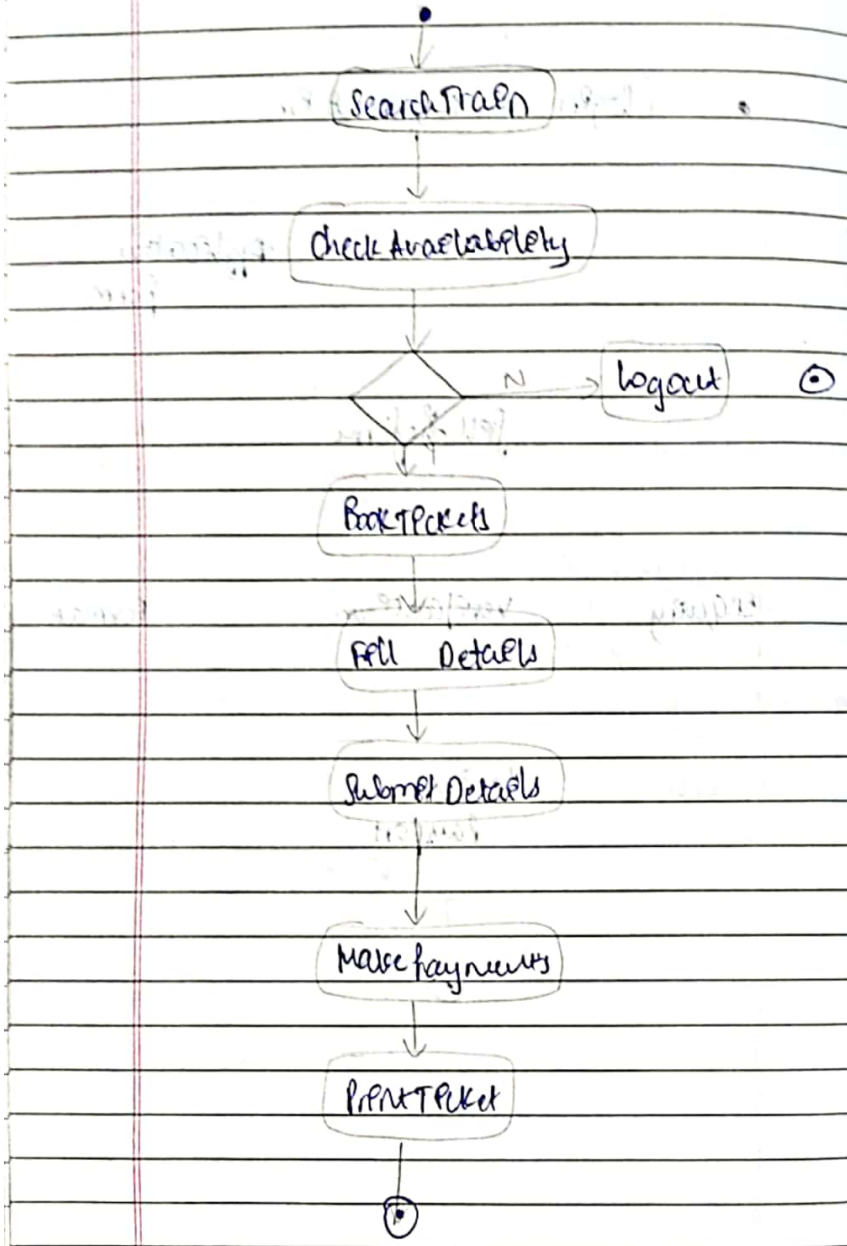
Stock Management System.



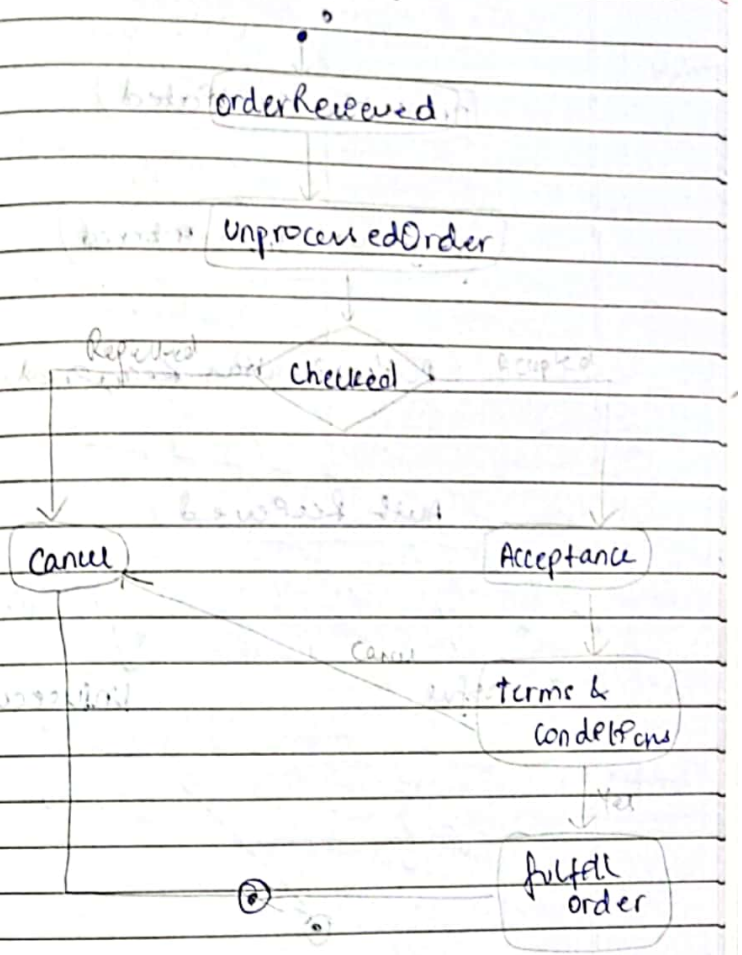
Passport Automation System



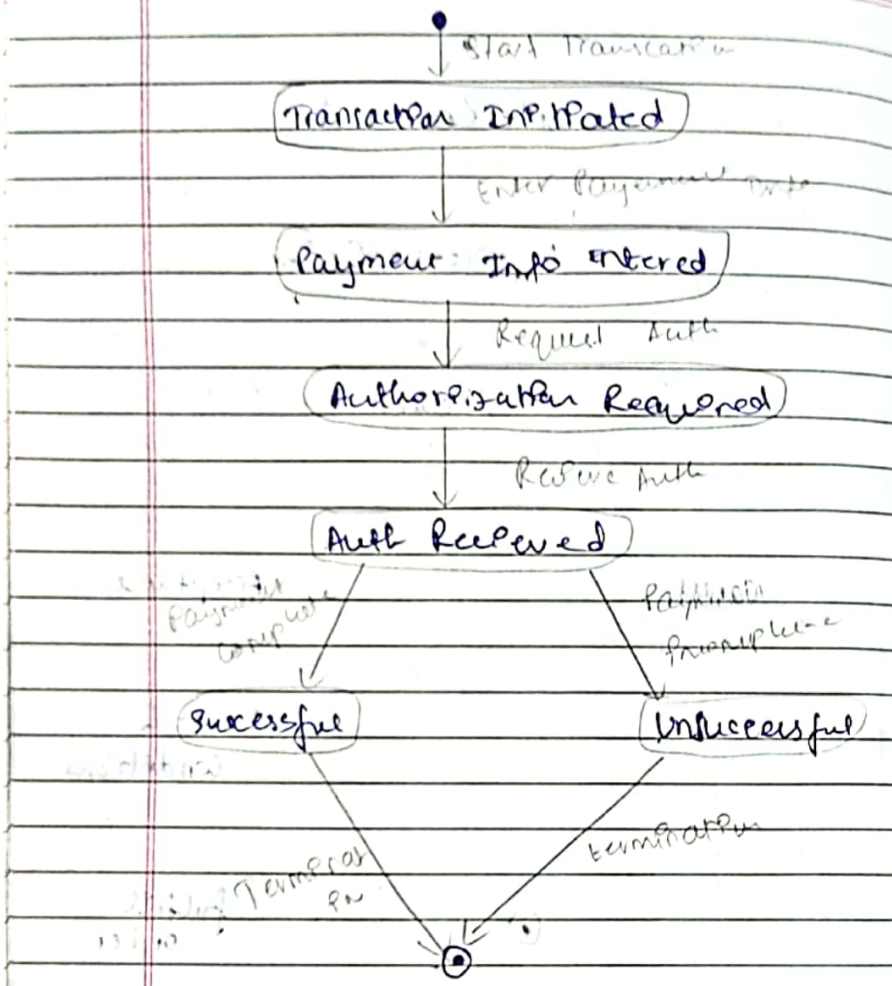
Railway Reservation System



Online Shopping System

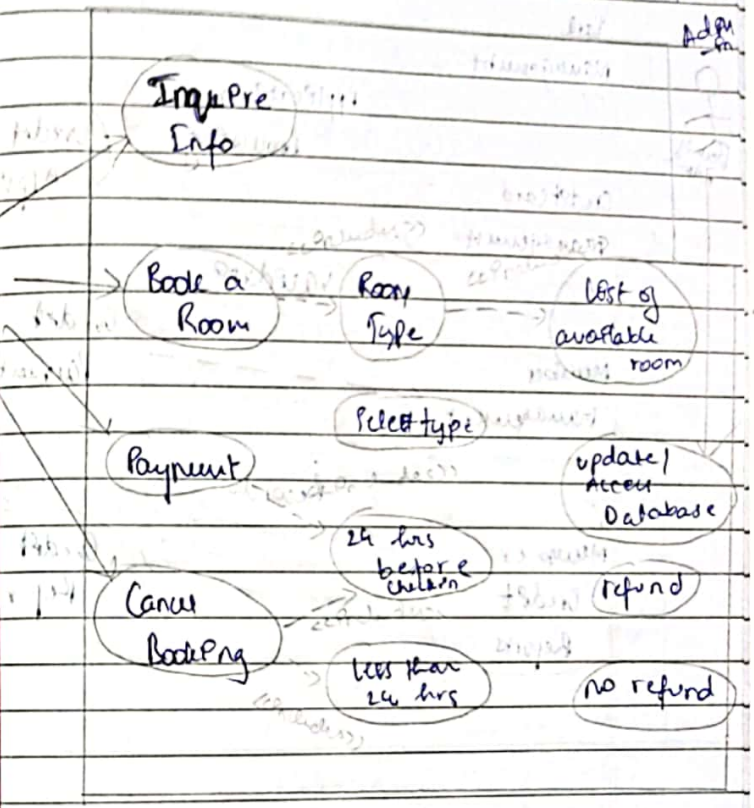


Credit Card system.

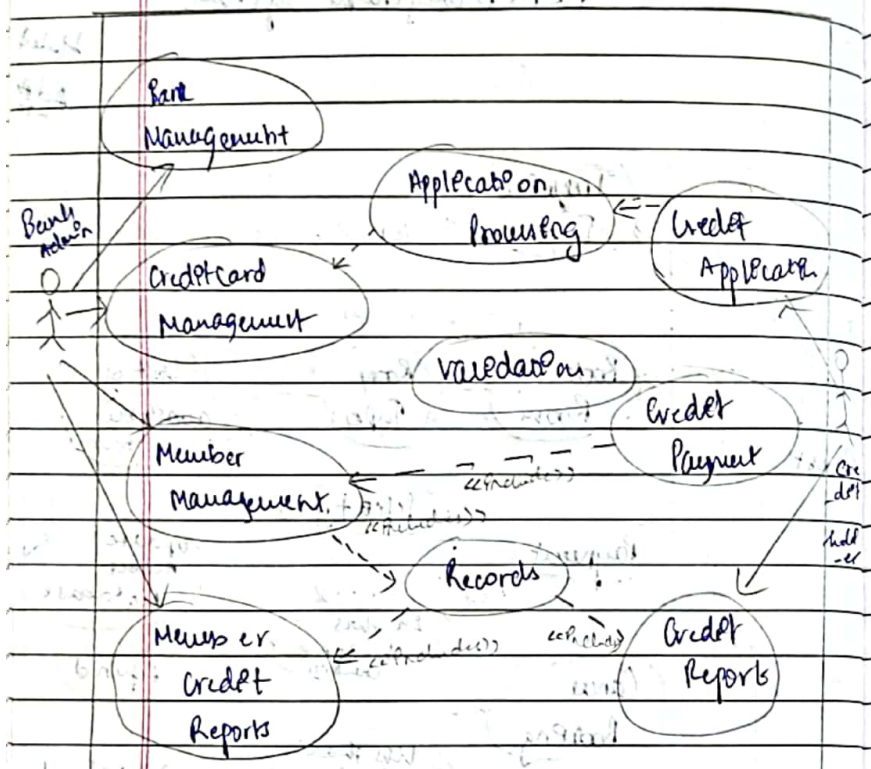


USE CASE DIAGRAMS

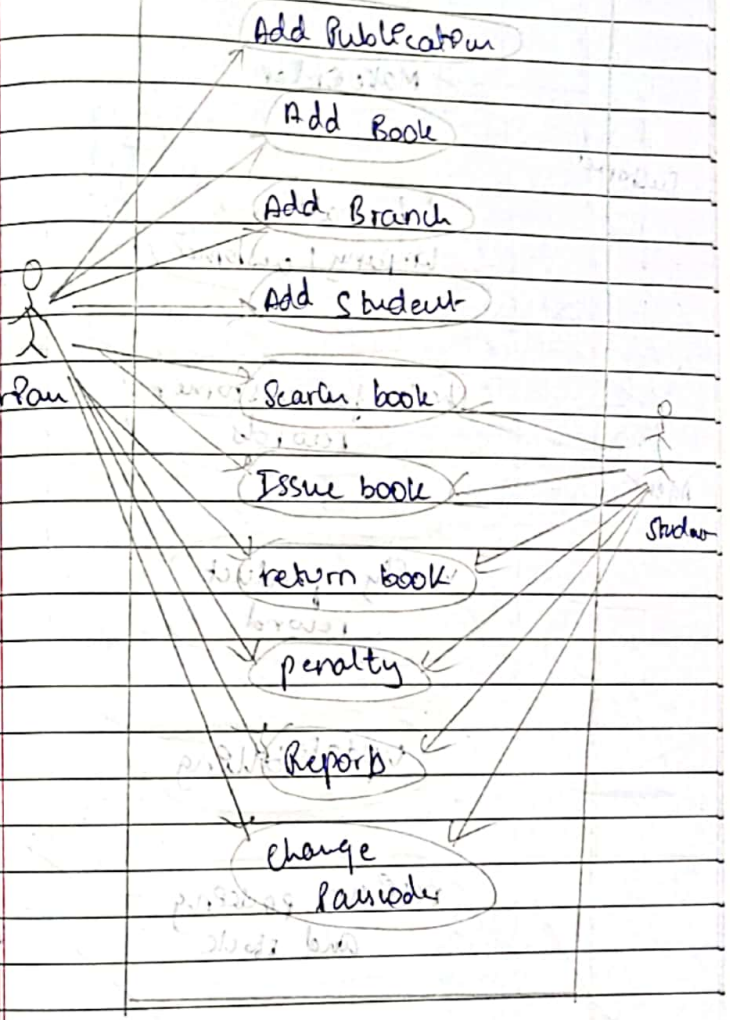
Hotel Management System.



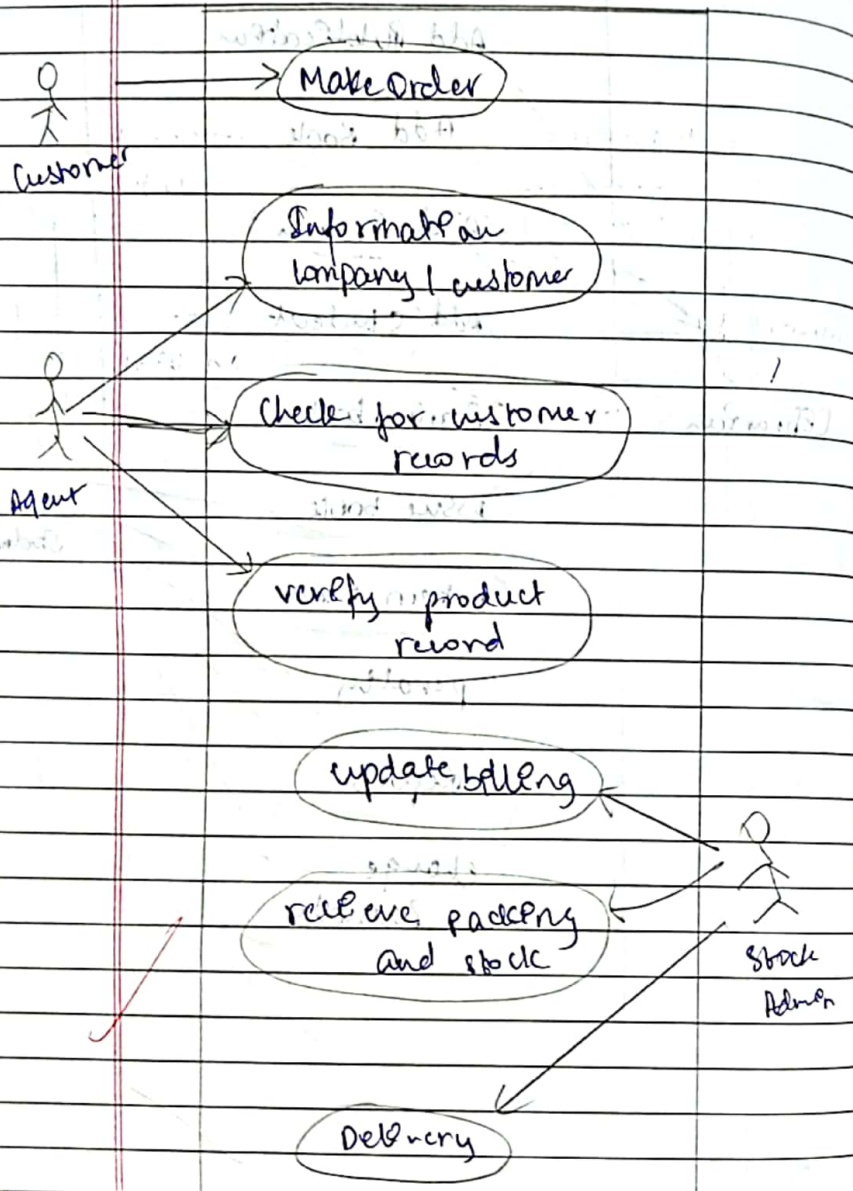
Credit Card Management system



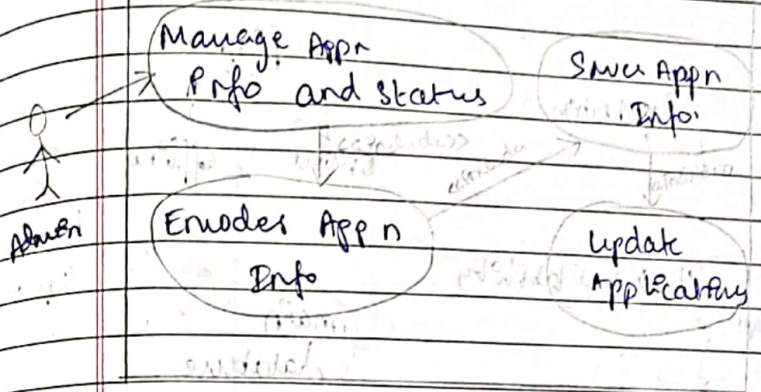
Library Management System



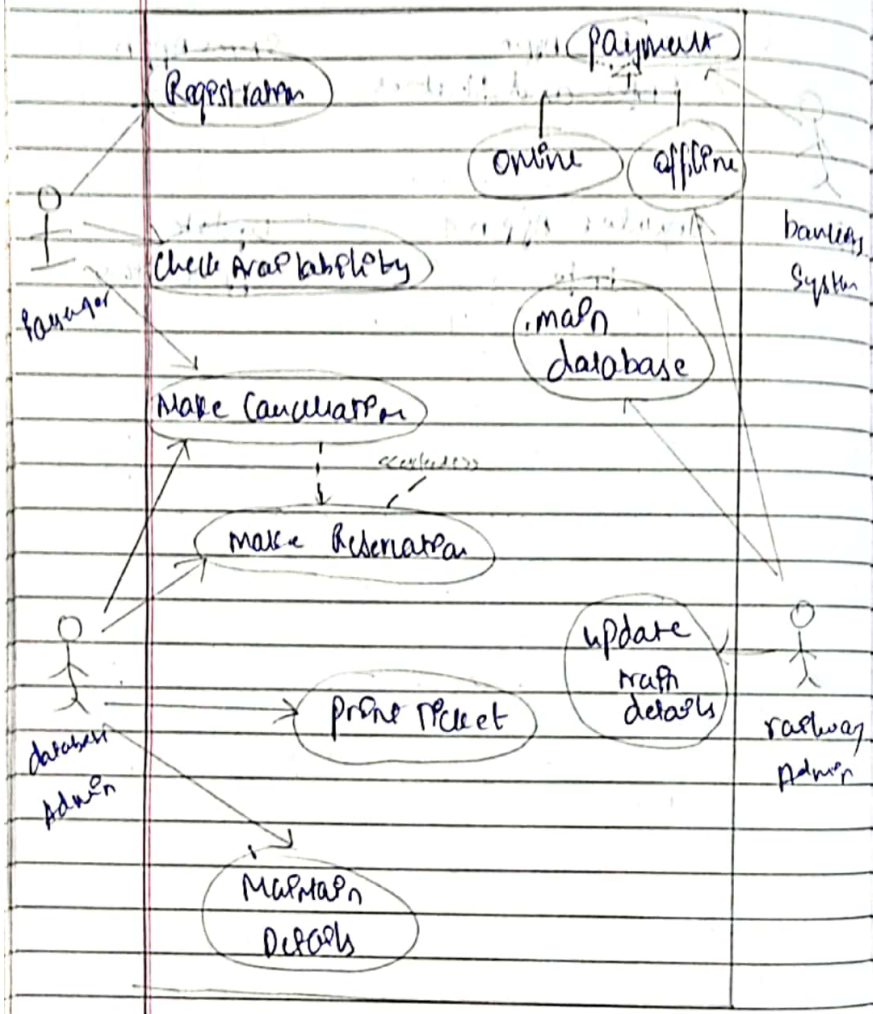
Stock Maintenance System.



Passport Automation System.

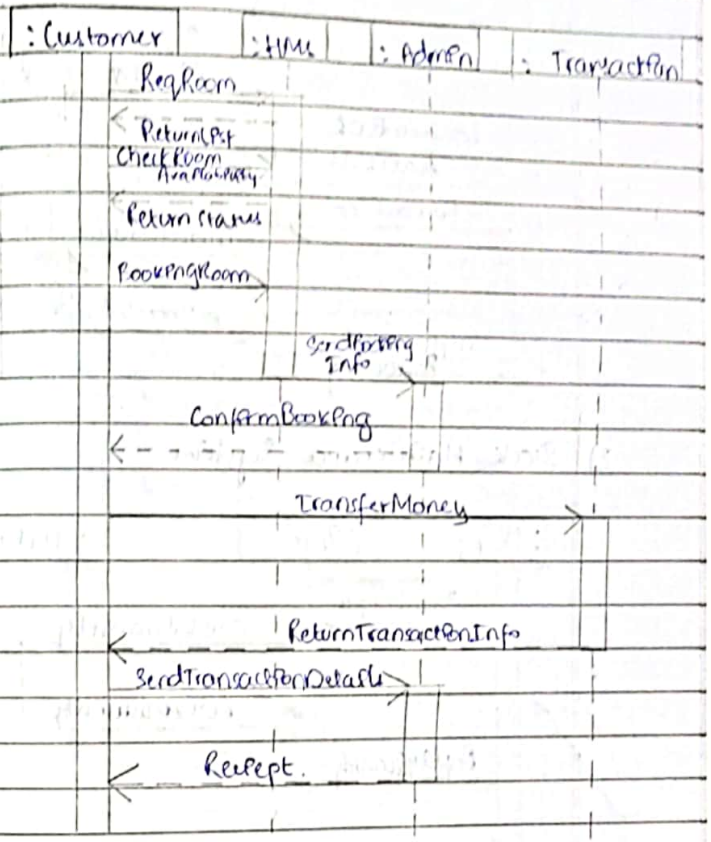


Railway Management System.

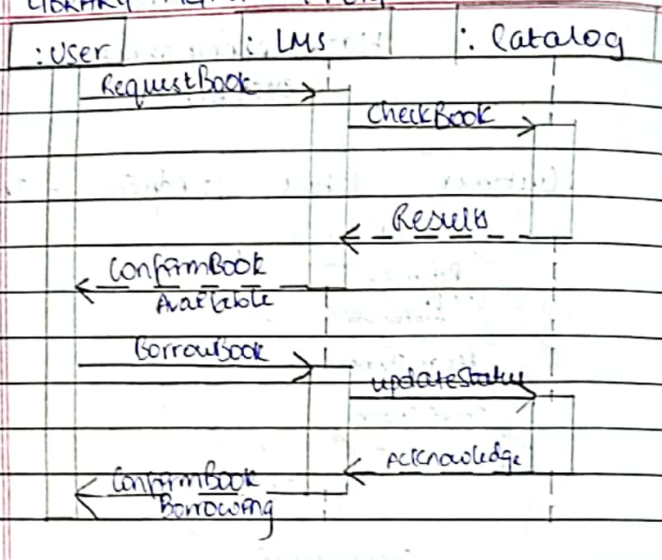


SEQUENCE DIAGRAMS

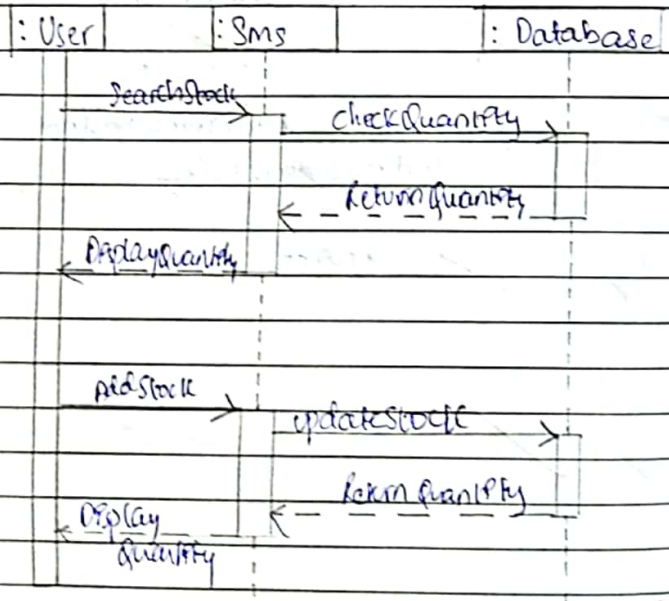
HOTEL MANAGEMENT SYSTEM



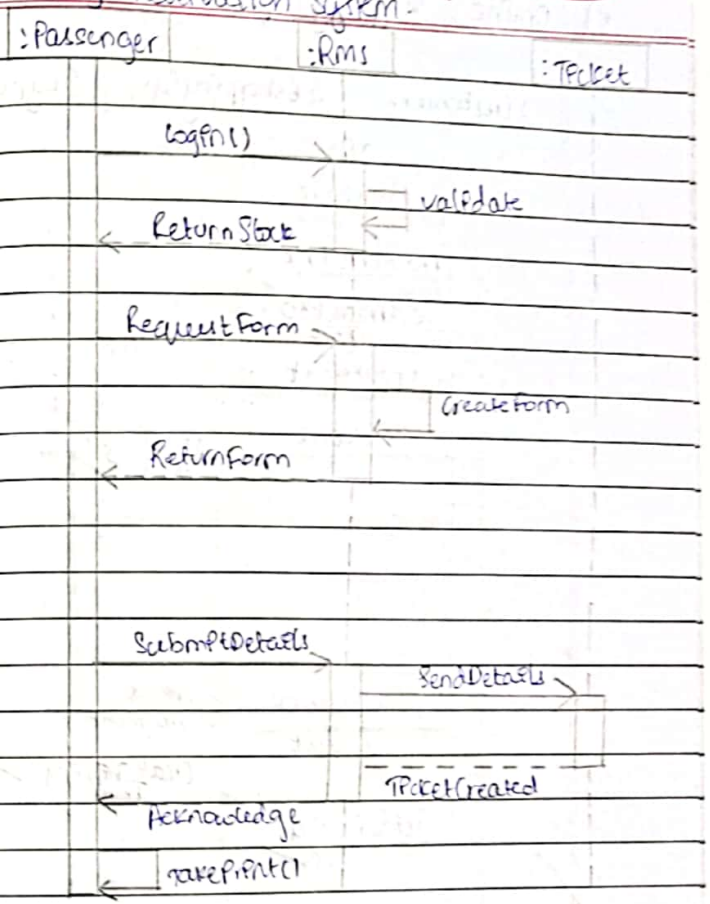
2) LIBRARY MGMT SYSTEM



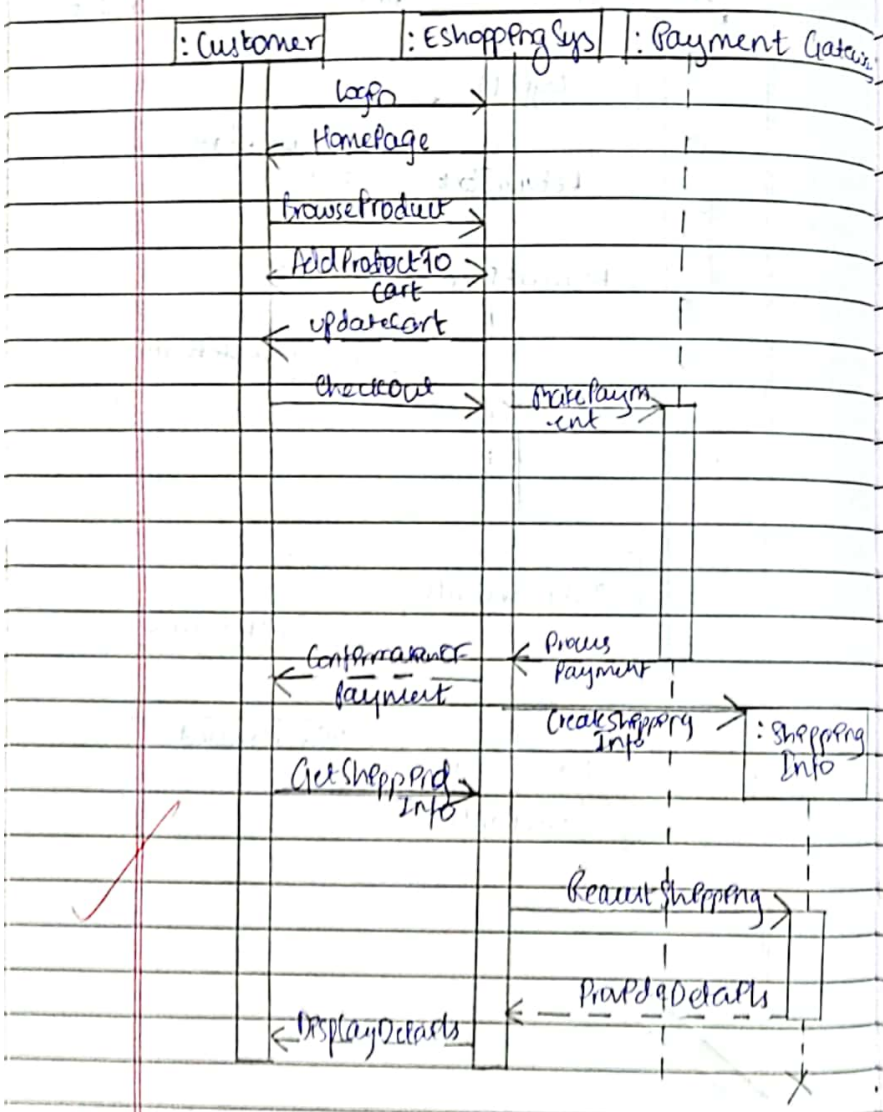
3) Stock Maintenance System



4) Railway Reservation System

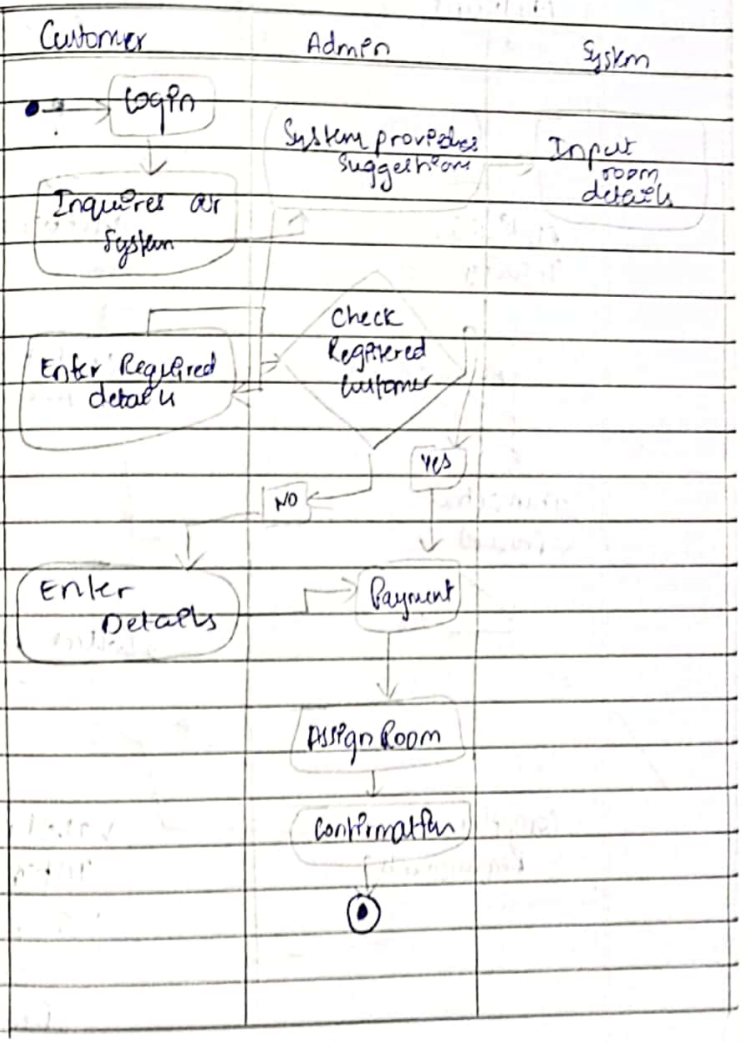


5) Online Shopping System.

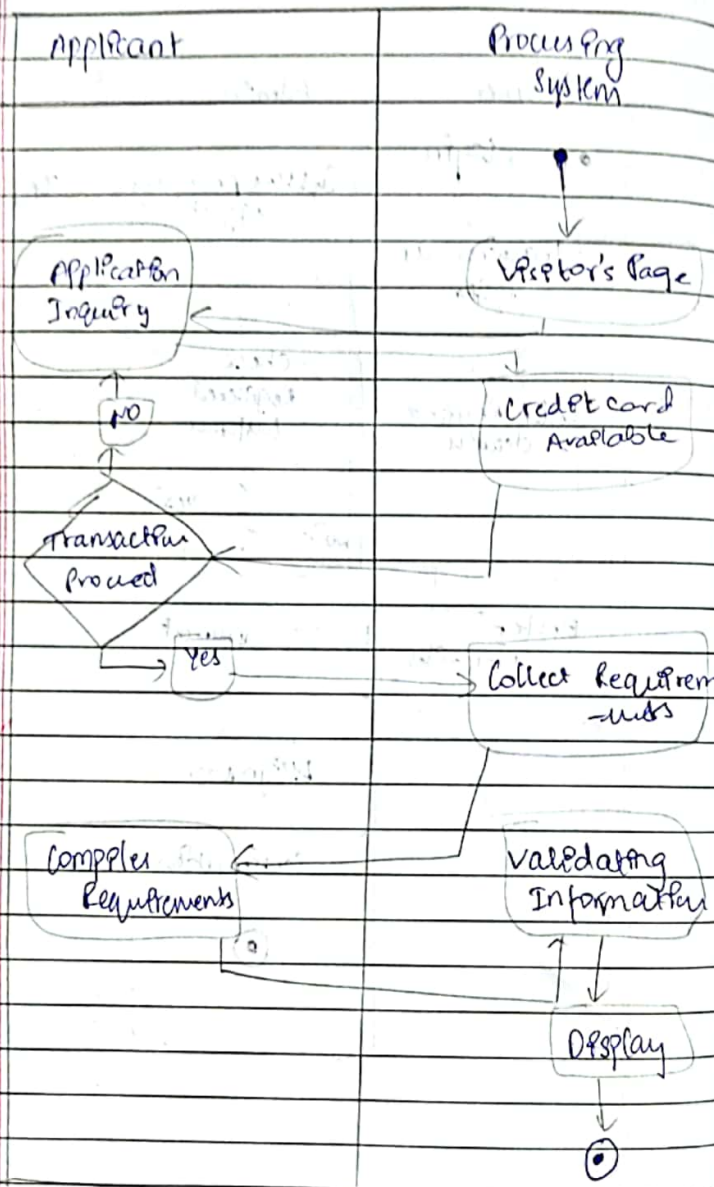


Activity diagrams:

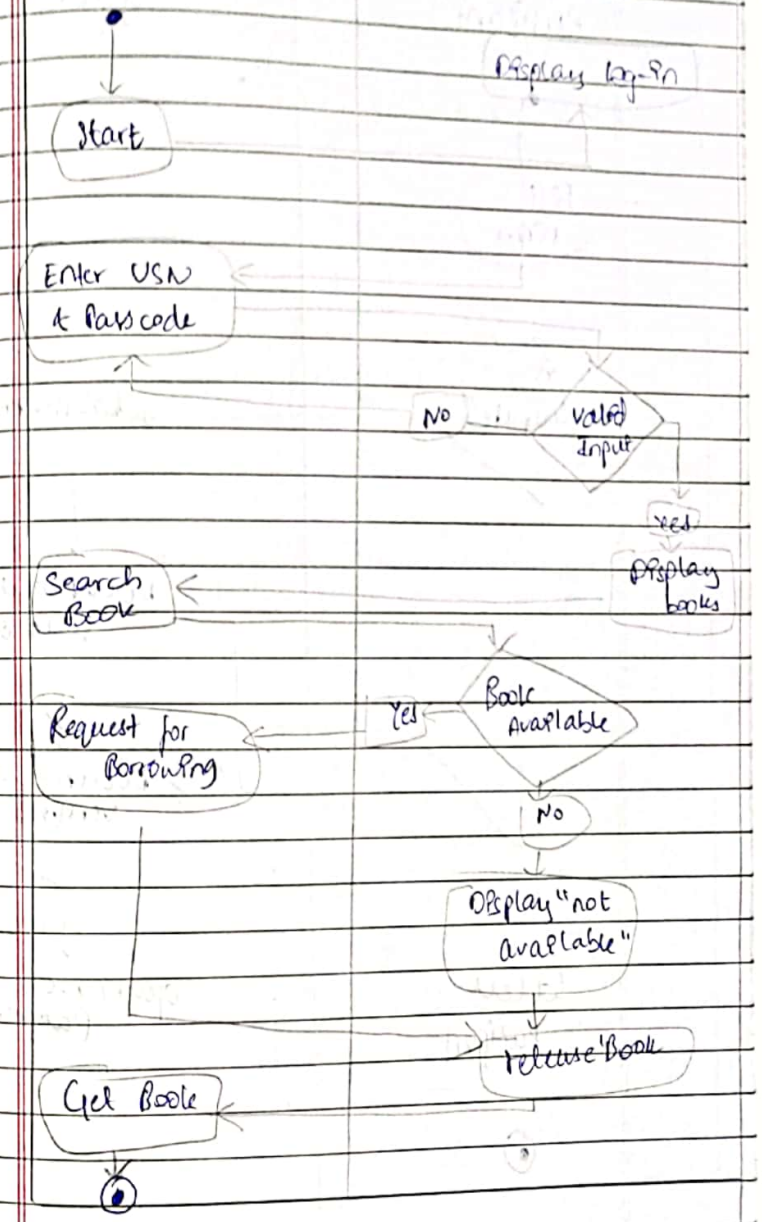
Hotel Management System.



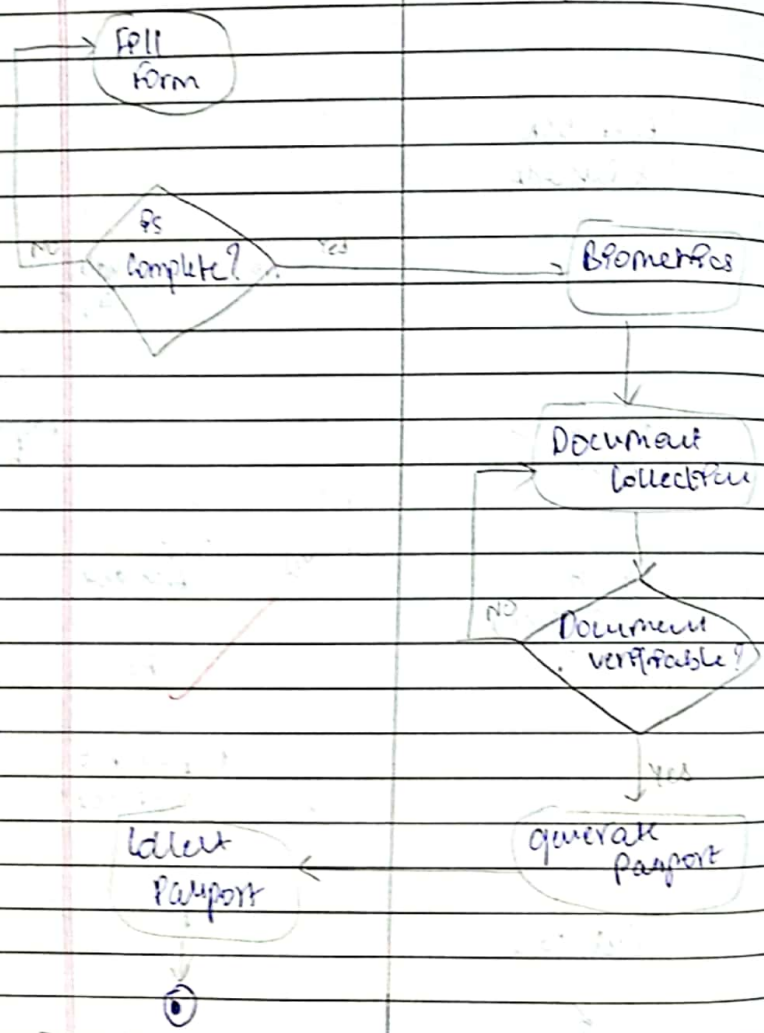
CMS



LMS



Passport System



Stock System