INDEX

NAME: Sohan A R STD.: ______ SEC.: ______ ROLL NO.: ______ SUB.: OOM LAB

30/9

Software Requirement Specifications (SRS)

1. Introduction

2 Purpose of this document

The purpose of this document is to outline the software requirements for the ~~Hotel~~ Credit card processing system. It serves as a comprehensive guide for stake holders, including developers, project managers, and clients, to understand the functionalities, features, and constraints of the system. The document will ensure that all parties have a clear understanding of the project objectives and deliverables.

3 Scope of this Document

The CCPS is designed to facilitate secure and efficient credit card transactions for businesses. It encompasses features such as transaction processing, fraud detection, reporting, and integration with various payment gateways. The estimated development cost is $200,000, with a timeline of 8 months of completion.

4 Overview

The credit card processing system will enable businesses to accept Credit card payments seamlessly. It will provide a secure interface for transaction processing, ensure compliance with financial regulations, and include features for reporting and fraud detection. The system will enhance user experience.

5 General Description:

The CCPS aims to streamline the credit card payment process for Businesses. key objectives include.

6 ~~Fundamental Requirements~~ User characteristics:

Merchants: Business owners and staff who will process transactions

Customer: Individuals making purchases using credit cards.

features:
User friendly interface for processing payments.
Support for various credit card brands and payment methods.

Benefits:
Increased sales through flexible payment options
Enhanced security and fraud prevention

Importance:
Providing a reliable credit card processing solution is critical for businesses to thrive in a competitive market.

Functional Requirements.
User Authentication. Secure login for merchants and administrators
Transaction processing: Ability to process credit card transactions in real time.
Refund management: Facilitate processing of refunds and charge backs
Fraud detection: Implement mechanisms to and prevent fraudulent transaction.

Reporting: Generate transaction reports, including sales summaries and dispute resolutions.
Integration: API for integration with e commerce plat and point of sale (POS) system

Interface Requirements
The CCPS will include the following interfaces

Merchant Interface:
A web based dashboard for merchants to manage trans
-ctions and access reports.

Customer Interface: A secure payment portal for customers to enter credit card details
API Interfaces: RESTful API's for third party ~~applications~~ integrations and payment gateway connections

5 Performance Requirements
The system should meet the following performance criteria

Response time: All transaction requests should be processed within 3 seconds

Concurrent Users: Support up to 500 concurrent transactions without performance issue.

6 Design Constraints
The design of the CCPS will adhere:

Technology Stack: Must utilize specified technologies
Compliance: Must adhere to Payment Card Industry data Security standards

7 Non Functional Attributes
Security: Implement system encryption protocols
Usability: Intuitive interface for ease of use by merchan
Portability: Compatible across multiple platforms

8 Preliminary Schedule and Budget:
The project is scheduled for a duration of 8 months, wit following milestones:

Requirements Gathering: 1 month
Design Phase: 2 months

elopment phase: 4 months
esting and Deployment: 1months
Budget Estimate
otal Estimated Cost: $200,000
Development: $120,000
Testing: $40,000
Project Management: $40,000

30/9

Date 7 Oct

LAB-2

HOTEL MANAGEMENT SYSTEM

1 INTRODUCTION

1.1 Purpose of this document:

The purpose of this document is to outline the software requirements for the hotel management system. It serves as a comprehensive guide for stake holders, including developers, project managers, and clients, to understand the functionalities features and constraints of the system

1.2 Scope of the document:

The Hotel Management System (HMS) is designed to cater all the residents of the hotel as well as its staff by managing the finances, delegation of staff, hospitalit management, Background verification of residats, payment processing of tariffs of rooms etc.

1.3 Overview:

The Hotel Management System (HMS) provides an one stop solution, user interface, by managing the payments, staff attendance, staff allotment, incentive schedule and over-time payout, it also manages the number of residats in the hotel rooms by acting as a bridge between the hotel staff and its residats.

2 General description:

The HMS aims to streamline the management protocols of the hotel industry. Its key objectives include:

User characteristics:

Customers: Individuals staying in hotels for vacation.

Hotel staff : Manage customers and maintain the hotel.

features :
User friendly Interface to manage the staff and customers of the hotel. And maintainence of the hotel.

3 Functional Requirements
(i) User registration and Authentication
ii) Reservation management
iii) check-in | check out process
iv) Billing management
v) Room Management.

4) Interface Requirements
(i) User interface (UI) : Provide a user friendly interface for the users to interact with
ii) Api's : The api's provide improved integration and fa development.
iii) Database Interface : A proper database interface allows the hotel managment to feed the related statistics onto the databa

5) Performance requirements
(i) Response time
ii) Concurrent users
iii) Data storage

6) Design constraints
i) Platform
ii) Database
iii) Framework

7) Non functional attributes
i) Security

ii) Scalability
iii) Reliability

~~8) Preliminary Schedule and Budget:~~
~~i) Estimated Development duration~~
ii) [illegible]

iii) [illegible]

8) Preliminary Schedule and Budget
i) Estimated Development duration : 1 month
ii) Estimated Budget : $340,000
iii) Testing and deployment : $100,000
iv) Project management : $150,000 rev. date
v) Development : $90,000

7/10/24

Software Requirement specification (SRS)

LIBRARY MANAGEMENT SYSTEM

1. INTRODUCTION

1.1 Purpose of this document :

The purpose of this document is to outline the software re-
-ments for the library management system. It serves as a comprehensive guide for its stake holders, including developers, project managers, and client, to underst the functionalities features and constraints of the sys

1.2 Scope of this document.

the library Management System (LMS) is designed to man and automate the library resources, including cataloging, us -er management, and transaction processing. This document details the project objectives, expected benefits, estimated costs and timelines ensuring value for the library and its us

1..3 Overview.

The library management system will serve as centralized platform for managing library operations, enhancing efficiently, improving user Experience, and providing tool for library staff to manage resources effectively

2 General Description

The LMS targets library staff and members, providing features such as book cataloging, membership management, lending and returning books, and generating reports. the systems primary objective is to simplify library operation, improve resource accessibility.

3 Functional Requirements

1) User registration and Authentication

ii) Book Cataloging

ii) Memberships management

iii) Book lending

4) Interface Requirement

i) User Interface (UI)

ii) API's

iii) Database Interface

5) Performance Requirements

i) Response Time

(ii) Concurrent users

(iii) Data storage

6) Design Constraints :

(i) Platforms

(ii) Database

(iii) Framework

7) Non functional Attributes

i) Security

ii) Reliability

iii) Scalability

8) Preliminary Schedule and Budget

i) Estimated Development duration : 12 months

ii) Estimated Budget : $200,000

iii) Testing and Deployment : $100,000

iv) Project management : $50,000

v) Development : $50,000

8

SRS document for Stock Maintainence System

1 INTRODUCTION

1.1 Purpose of this document

This document outlines the functional and non-functional requirements for the Stock Maintainence System. It serves as a reference for developers, testers and stake holders, ensuring alignment on the system's goal.

1.2 Scope of the Document

The Stock Maintainence System is designed to automate the tracking and management for businesses, enabling users to monitor stock levels, manage product orders, deliveries and sales and generate detailed reports on inventory status

1.3 Overview

The stock Maintainence System will provide real time updates on stock levels and automate key functions such as low stock alerts, purchase order management, and sales tracking. It will generate reports to help businesses make data driven decisions regarding inventory management.

2 General Description

2. The Stock Maintainence System is a standalone application that can be integrated with existing Point-of-Sale (P systems and e-commerce platforms. It is designed to provide inventory management capabilities for businesses of various si with a focus on simplifying stock tracking.

3 Functional Requirements

3.1 stock management:

- The system should allow users to add, update and delete stock items and should provide Real time stock level updates.

3.2 Order management

- The system should generate and manage purchase orders for suppliers, so that users can track the status of their orders

3.3 Sales Management

- The system will reduce stock quantities automatically after each sale

3.4 User management : Each user must have a unique login and actions must be logged

4 Interface Requirements

4.1 User Interface => web based interface accessible through modern browsers

4.2 System Interface => System must support integration with third party POS and ERP Systems via API'S

5 Performance Requirements

- System must update the stock levels after 3 seconds of sales trade
- The system must not exceed the error rate of 0.1%

6 Design Constraints

- Use MySQL database, React JS for front end, and NodeJS for backend
- Optimize for use on mobile devices

7 Non Functional Attributes

- Security
- Portability
- Reliability
- Usability

Preliminary Schedule

Phase	Description	Duration
Requirements and Planning	Collect requirements and plan the project	2 weeks
System Design	Design the architecture and Interface	3 weeks
Development	Develop core features and functionalities	8 weeks
Testing and Debugging	Perform testing and fix bugs	3 weeks
Deployment and final review	Deploy and finalize	1 week
Total Duration		17 weeks

Budget

Item	Cost Estimate
Development cost	$30,000
Testing Cost.	$10,000
Deployment costs	$5,000
Maintenance Costs	$5000 per year.
Total Estimated Budget	$45,000.

SRS document for Passport Authentication System

1 Introduction

1.1 Purpose of this Document

The purpose of this document is to dyse the functional and non functional requirements for a passport Authentication System. It serves as a datated guide for developers, testers and stake holders to ensure a clear understanding of the systems goal, scope and features.

1.2 Scope of the System

The Passport Authentication System will provide a secure method for verifying the authenticity of passports by cross referencing the passport data with a global or national database. The system will reduce the risk of passport fraud, improve border security, and streamline the process of identifying and verifying individuals

1.3 Overview

The Passport Authentication System is a software solution designed to authenticate passports in real time by scanning and validating passports against a secure, centralized database. The system will now allow authorized users to scan a passport, verify its credibility and validate it instantly.

2 General Description

2.1 The Passport Authentication System will function as a standalone application, but will also be capable of integration with existing border security and immigration systems. It will be designed to ensure fast and accurate authentication of passports at border-checkpoints, airports and other secure locations.

3 Ponctional Requirements

- Passport Scanning and Data capture
  The system must capture from the MRZ i.e. machin
  readable zone using a compatible scanner
- Passport Data Validation
- Fraud Detection
- Reporting and logs.

4 Interface Requirements

- User Interface
- System Interface

5 Performance Requirements.

- Must authenticate Passport in 3-5 seconds
- System must maintain accuracy rate of over 99.9%

6 Design Constraints

- The system should be built using a robust and secure database.
- It must comply with the International Security standards

7 Non Functional Attributes

- Security
- Portability
- Reliability
- Usability
- Scalability

Preliminary Schedules and Budget

	Phase	Description	Duration
1	Requirements gathering and Planning	Collect system requirements and plan the project	3 weeks
2	System Design	Design the architecture and User interface	4 weeks
3	Development	Develop Core functionalities and System Integration	10 weeks
4	Testing and De bugging	Perform System testing and bug fixe	4 weeks
5.	Deployment and final Review	Deploy the system and review final deliverables	2 weeks
	Total Duration		23 weeks

Preliminary Budget

Item	Cost Estimate
Development Costs	$ 50,000
Testing Costs	$ 15,000
Deployment Costs	$ 10,000
Maintenance Costs (Yearly)	$ 5,000
Total Estimated Budget	$ 75,000

SRS Document for Railway Reservation System...

1 Introduction

1.1 Purpose of this Document

The purpose of this document is to outline the functional and non functional requirements for the Railway Reservation System. This document will serve as a guide for developers, testers and stake holders to ensure a clear understanding of the systems goals and objectives

1.2 Scope of the System

The RRS is intended to automate the railway ticketing and reservation process, allowing passengers to book tickets online, check seat availability, and view train schedules. The system will reduce the manual effort involved in ticket reservations and improve customer Satisfaction.

1.3 Overview

The RRS will be a web based platform that provides passengers with the ability to search for trains, book tickets, cancel reservations, and manage their travel itineraries.

2 General Description

The Railway Reservation System will operate as an indep online platform that can integrate with existing railway databases for seat allocation, ticket booking and schedule management. It provides real time updates on train schedules, seat availability and booking statuses

3 Functional Requirements

- User Registration and Login
- Ticket Booking

- Reservation Management
- Schedule Information
- Reporting and logs
- Administrator functions

4 Interface Requirements.
- User Interface
- System Interface

5 Performance Requirements
- The System Must be able to handle up to 5000 concurrent users without significant performance degradation
- The system must process a booking within 10 seconds including payment confirmation

6 Design Constraints
- The system must comply with data protection regulations such as GDPR to protect users personal payment information.
-
- The system will be built using a relational database to manage ticket bookings, user data and train schedules

Non Functional Attributes
- Security
- Portability
- Reliability
- Usability
- Scalability

Preliminary Schedule and Budget

=>

Phase	Description	Duration
Requirements Gathering and Planning	Collect system requirements and plan the proj	3 weeks
System Design	Design the system arc and user interface	4 weeks
Development	Develop core functionalities including booking and payment integration	10 weeks
Testing and Debugging	Conduct system testing	4 weeks
Deployment and final Review	Deploy the system, perform user training and conduct final review	2 weeks
Total Duration		23 weeks

Preliminary Budget

Item	Cost Estimate
Development Costs	$ 60,000
Testing Costs	$ 12,000
Deployment Costs	$ 8,000
Maintenance Costs	$ 7000
Total Estimate	$ 80,000

SRS document for Online Shopping System,

1 Introduction

1.1 Purpose of this document

The purpose of this software Requirement Specification (SRS) is to outline the functional and non functional requirements for the Online Shopping System. This document will serve as a guide for developers, testers and stake holders to ensure a clear understanding of the system goals and objectives

1.2 Scope of this Document

This document covers the functional and non-functional requirements for the Online Shopping System, including user interfaces, system interaction, and performance metrics. It provide a detailed overview of the system's capabilites.

1.3 Overview

The OSS will be a online web based platform that allows users to browse products, add items to their cart, manage their users to browse products, add items to their cart, manage their user accounts and complete purchases using various payment methods.

2 General Description

2.1 Product Perspective

The Online Shopping System will function as a standalone application but may integrate with third-party payment gateways and inventory management . Systems.

3 Functional Requirements.

- User Registration and LOGIN
- Product Management
- Shopping cart Operations

- Checkout process
- Order tracking

4 Interface Requirements
- User Interface
- System Interface

5 Performance Requirements
- The system should respond to user action within 2 seconds
- The system must support atleast 500 concurrent users
- The system must handle upto 10,000 product listings

6 Design Constraints
- Technology Stack
- Compliance

7 Non Functional attributes
- Security => User data must be encrypted
- Portability -> The application should be accessible from various devices and browsers
- Reliability => The system should maintain an uptime of 99

Preliminary Schedule and Budget

Phase	Description	Duration
Requirement Gathering	Collect System Requirements	2 weeks
System Design	Design the system architecture and user interface	4 weeks

Development	Develop core functionality Including booking and payment Integration	3 weeks
Testing ad Debugging	Conduct system testing	2 weeks
Deployment and final review	Deploy the system perform user training	2 weeks
Total Duration		13 weeks

Preliminary Budget

Category	Estimated cost
Development	$ 100,000
Testing	$ 20,000
Project management	$ 15,000
Infrastructure	$ 10,000
Miscellaneous	$ 5,000
Total Estimated Budget	$ 15,000

Identify minimum 5 classes for each application:

Association Names

Association end Names

{bags} {sequences} {ordered}

Multiplicity

association class

Hotel Management System

Hotels : Represents the hotel entity, including its na
location, and amenities.

Room : Manages room details such as room number, t
/capacity and price

Customer : Represents the guest or customers, personal in
-tion and preferences

Staff : manages information about the hotel staff, inclu
roles schedule, and contact details

Payment : Deals with payment processing including methods
transaction details and invoicing

Booking : Handles booking information including customer details, c
in and check out dates and payment status

Association names

Hotel - Room

Association Name : Contains

Customer - Booking

Association Name : Creates

(3) Booking - Room
Association Name - reserves

(4) Payment - Booking
Association Name - covers

Association End Names

(1) Hotel - Room
Hotel : has
Room : belongs to

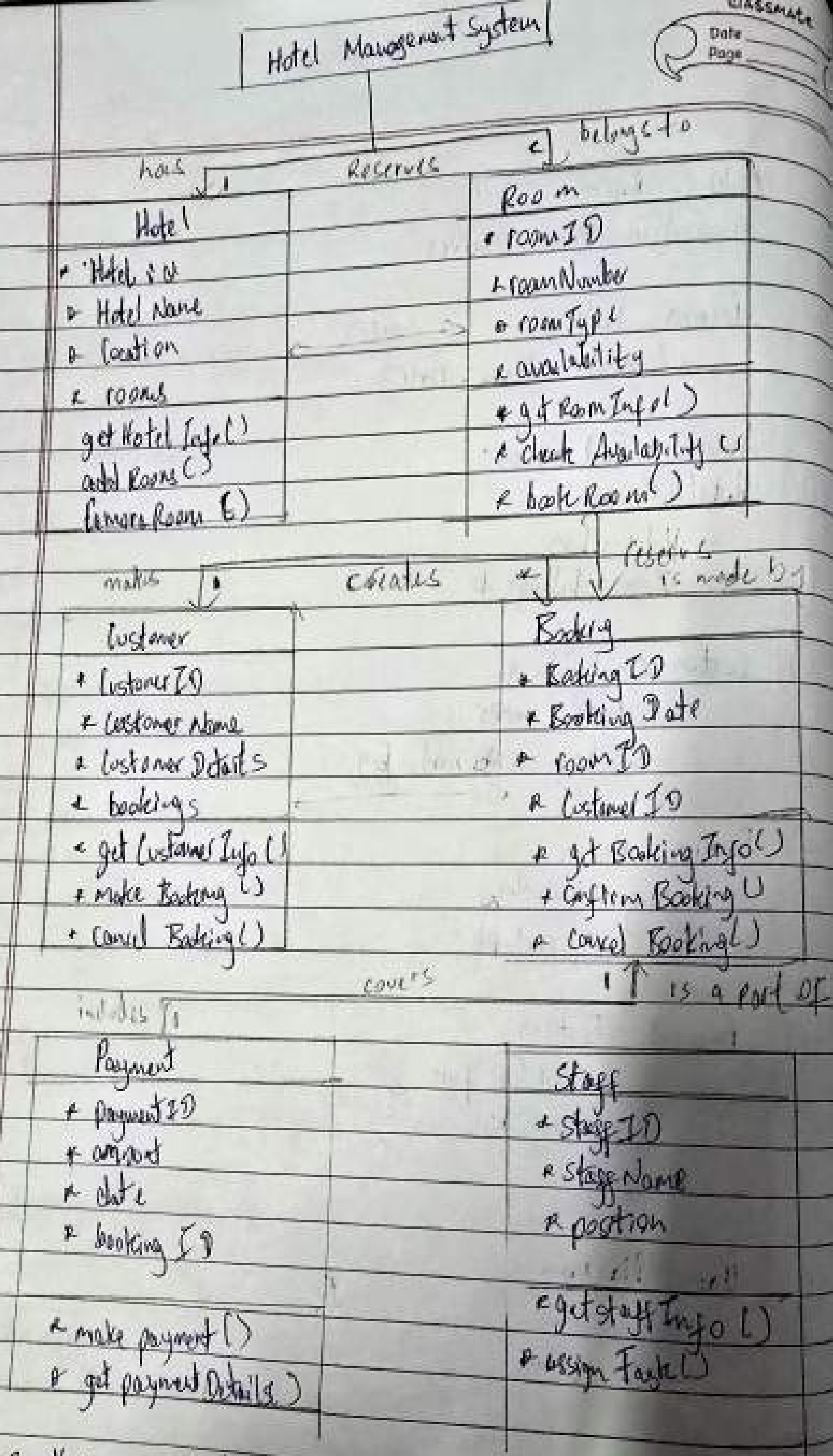
(2) Customer - booking
Customer : makes
Boo king : is made by

(3) Booking - Room
Booking : includes
Room : is part of

(4) Payment - Booking
Payment : is for
Booking : has

Class Diagram

=>



In this generalization we have Hotel as the super class and its rooms as the subclass being 1 to many can be ~~observed~~ between the classes

Credit card processing

Classes :

1 Credit card : Represents a credit card with unique details like card number, expiration date and card holders name

2 Transaction
Represents a transaction made ~~[illegible]~~ using a credit card, including transaction ID, date, amount, and merchant ID

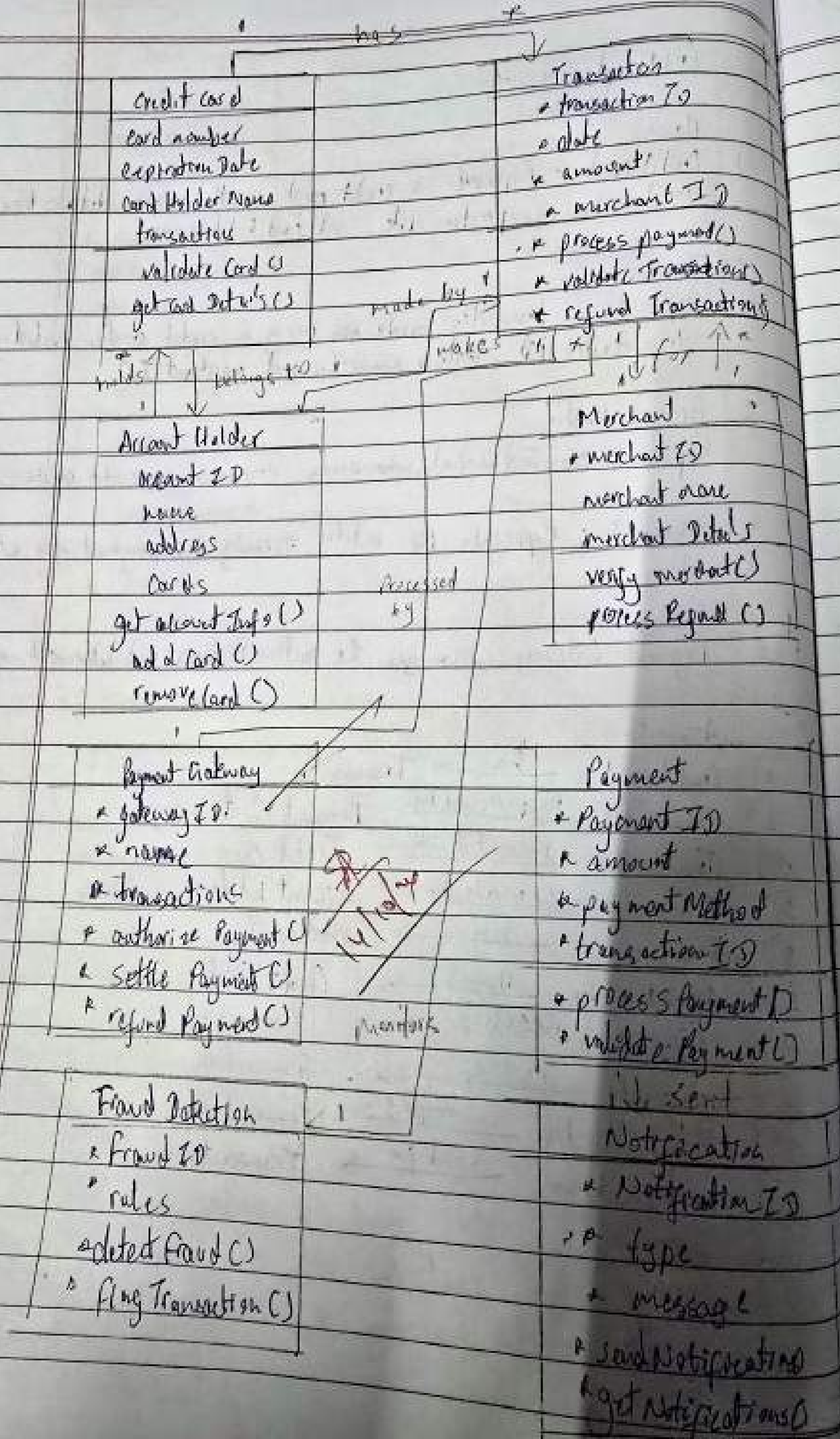
3 Account Holder :
Represents the individual who owns one or more credit cards

4 Merchant : Represents the entity acceping the payment for a transaction.

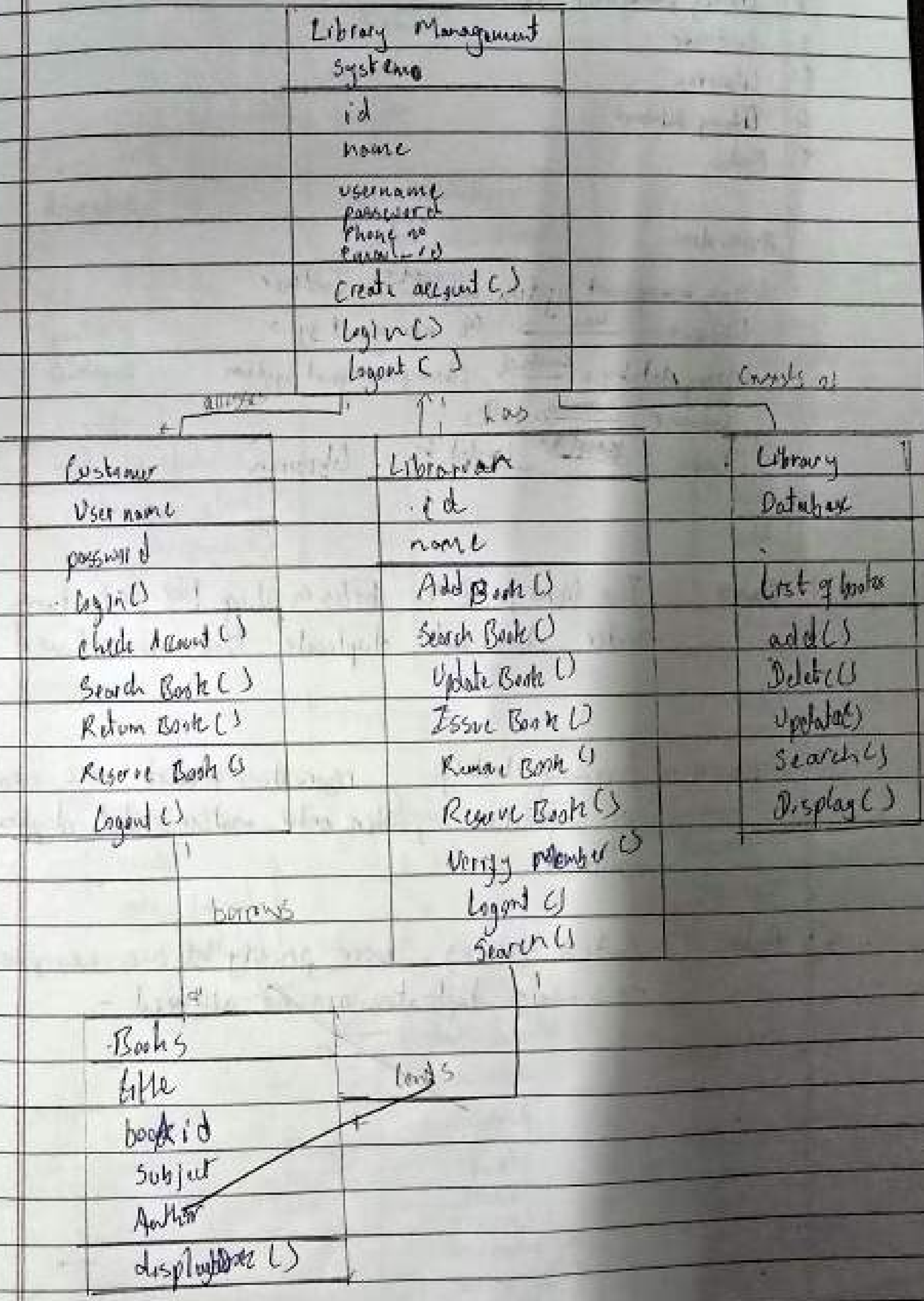
5 Payment Gateway : Manages the authorization and settlement of paym

Associations

1) Credit card $\xrightarrow{\text{has}}$ Transaction
2) Transaction $\xrightarrow{\text{is processed by}}$ Payment Gateway
3) Transaction $\xrightarrow{\text{associated with}}$ Credit card
4) Transaction $\xrightarrow{\text{is made by}}$ Account holder
5) Transaction $\xrightarrow{\text{by}}$ merchant
6) Account holder $\xrightarrow{\text{Owns}}$ Credit Card
7) Merchant $\xrightarrow{\text{Receives}}$ Transaction
8) Payment $\xrightarrow{\text{part of}}$ Transaction
9) Fraud Detection $\xrightarrow{\text{monitors}}$ Transaction
10) Notification $\xrightarrow{\text{sent for}}$ Transaction


has
Credit card
card number
expiration Date
card Holder Name
transactions
validate Card ()
get card Details ()
holds
belongs to
made by
makes
Transaction
* transaction ID
* date
* amount
* merchant ID
* process payment()
* validate Transaction()
* refund Transaction()
for
Account Holder
account ID
name
address
cards
get account Info ()
add card ()
remove card ()
Processed by
Merchant
* merchant ID
merchant name
merchant Details
verify merchant()
process Refund ()
Payment Gateway
* gateway ID
* name
* transactions
* authorise Payment ()
* settle Payment ()
* refund Payment ()
monitors
Payment
* Payment ID
* amount
* payment Method
* transaction ID
* process Payment ()
* validate Payment ()
sent
Fraud Detection
* Fraud ID
* rules
* detect Fraud ()
* flag Transaction ()
Notification
* Notification ID
* type
* message
* send Notification
* get Notifications ()

3 LIBRARY MANAGEMENT SYSTEM



Classes

1) Library Management System
2 Customer
3 Librarian
4 Library Database
5 Books

Associations

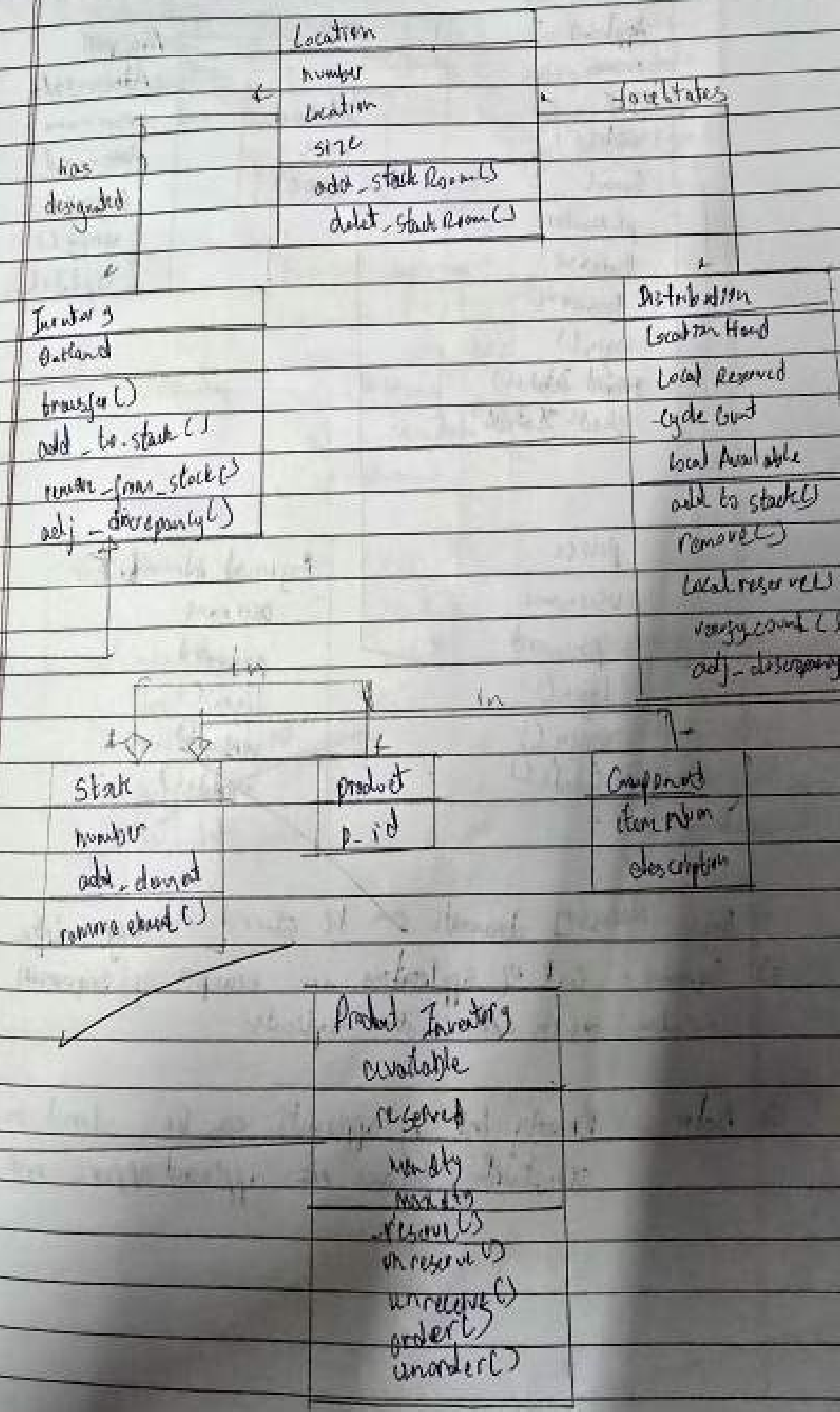
Library management system $\xrightarrow{\text{accessed by}}$ Customer
Librarian $\xrightarrow{\text{has}}$ lib management sys
Library database $\xrightarrow{\text{consists of}}$ Library management system
Customer $\xrightarrow{\text{borrows}}$ books
books $\xrightarrow{\text{loaded by}}$ Librarian

1) Bags : The library store books in bags like structures ha
order does not duplicate books are allowed

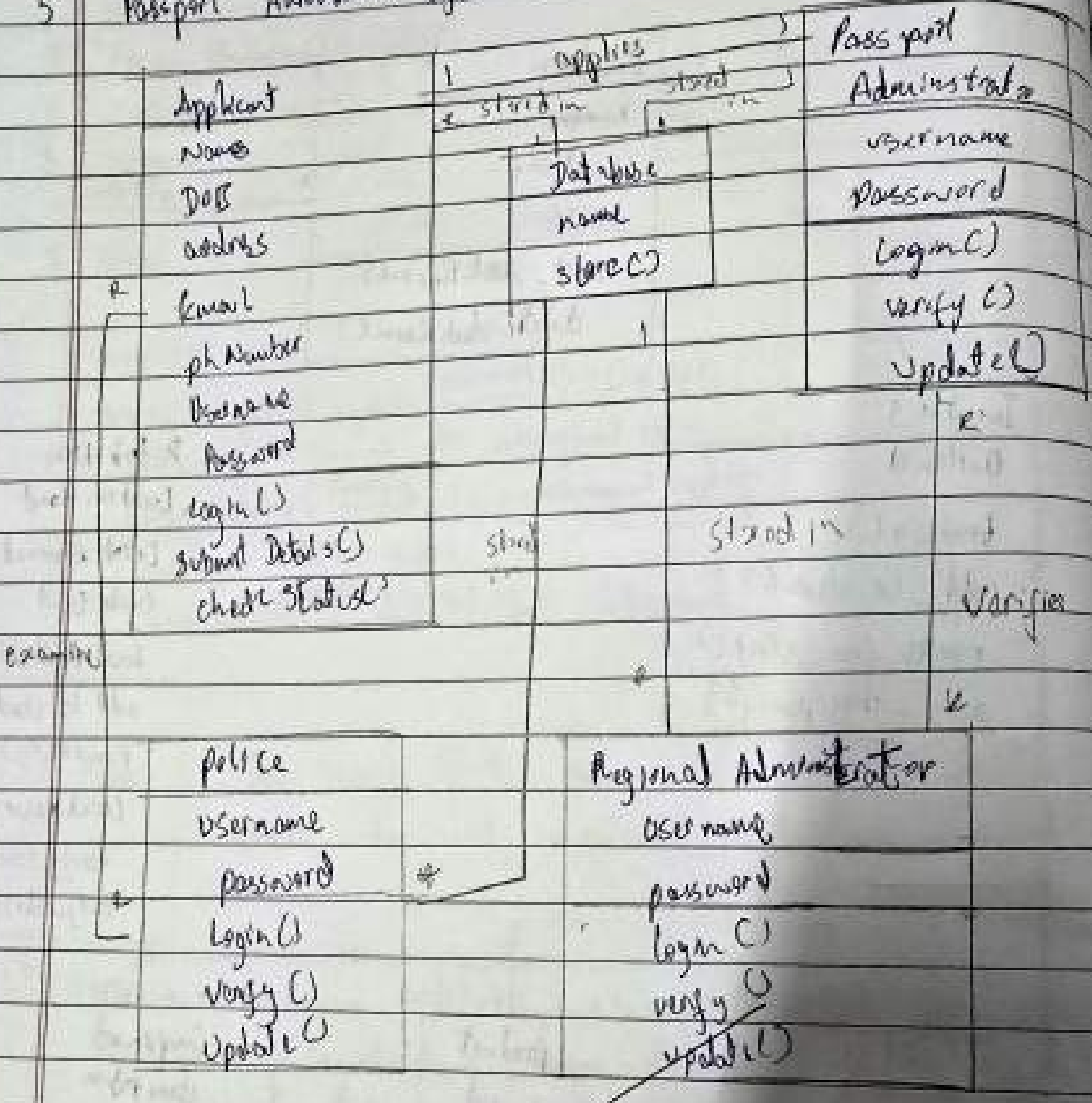
2) Sequences : Borrowing history , reservation queues are example
of sequences , where order matters but duplicates
allowed.

3) Orders : Sorted catalog, user priority list are examples of
-rs where duplicates are not allowed -

4 Stock Maintainance System



5 Passport Automation System



1) Bags - Necessary documents can be stored in bag like stru

2) Sequences - List of applications are examples of sequences where
-cations are in order with duplicates

3) Order:- Priority list of applicants can be stored in order
structure because one applicant appears one time

Applicant
name : string
DOB : number
photo : photo
gender : string

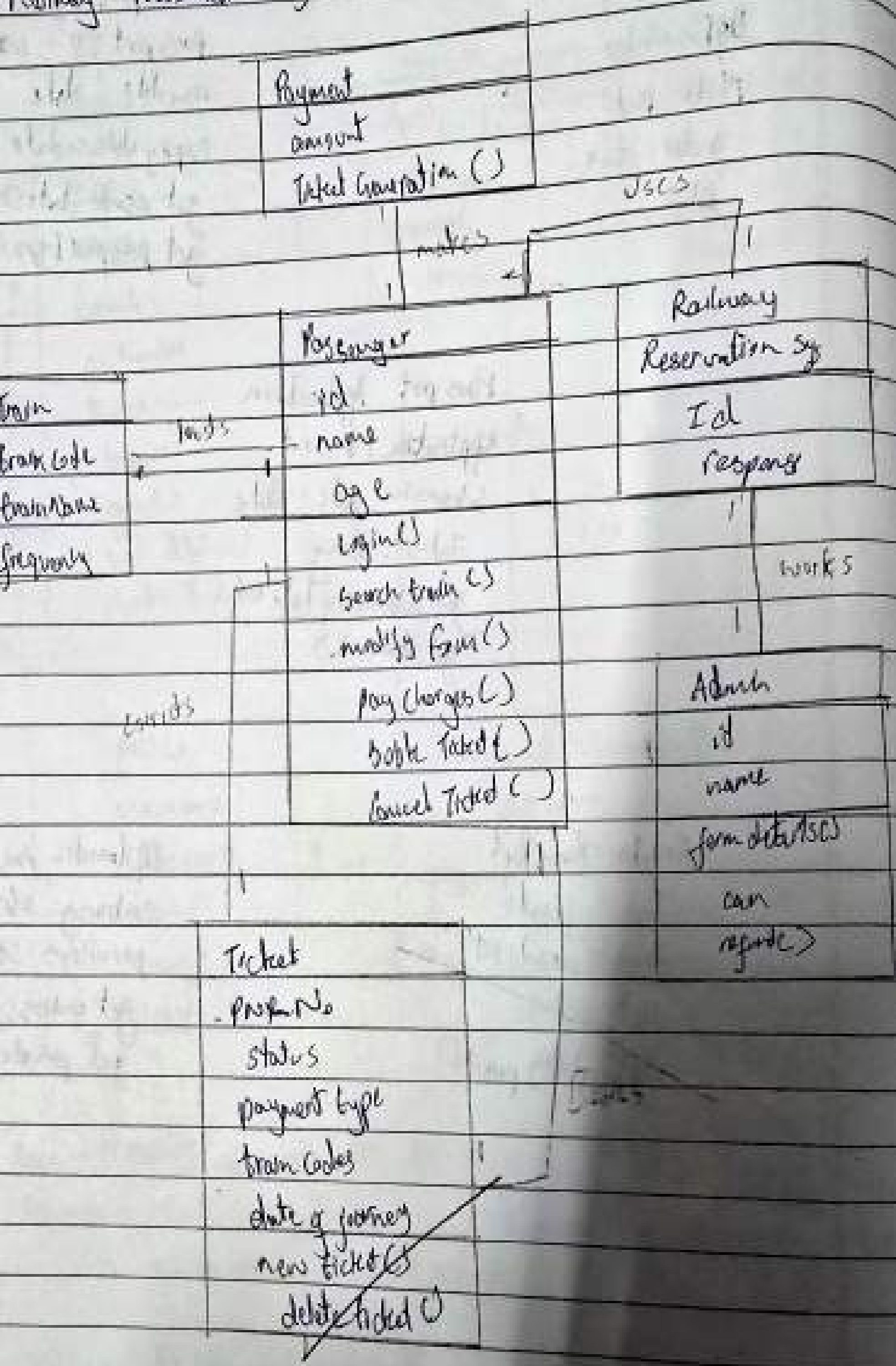
Passport
passport ID : int
issuedate : date
expiry date : date
get issude date()
get passport type()

Passport Application
application id : int
submission_date : date
status : string
get submissiondate()
getstatus()

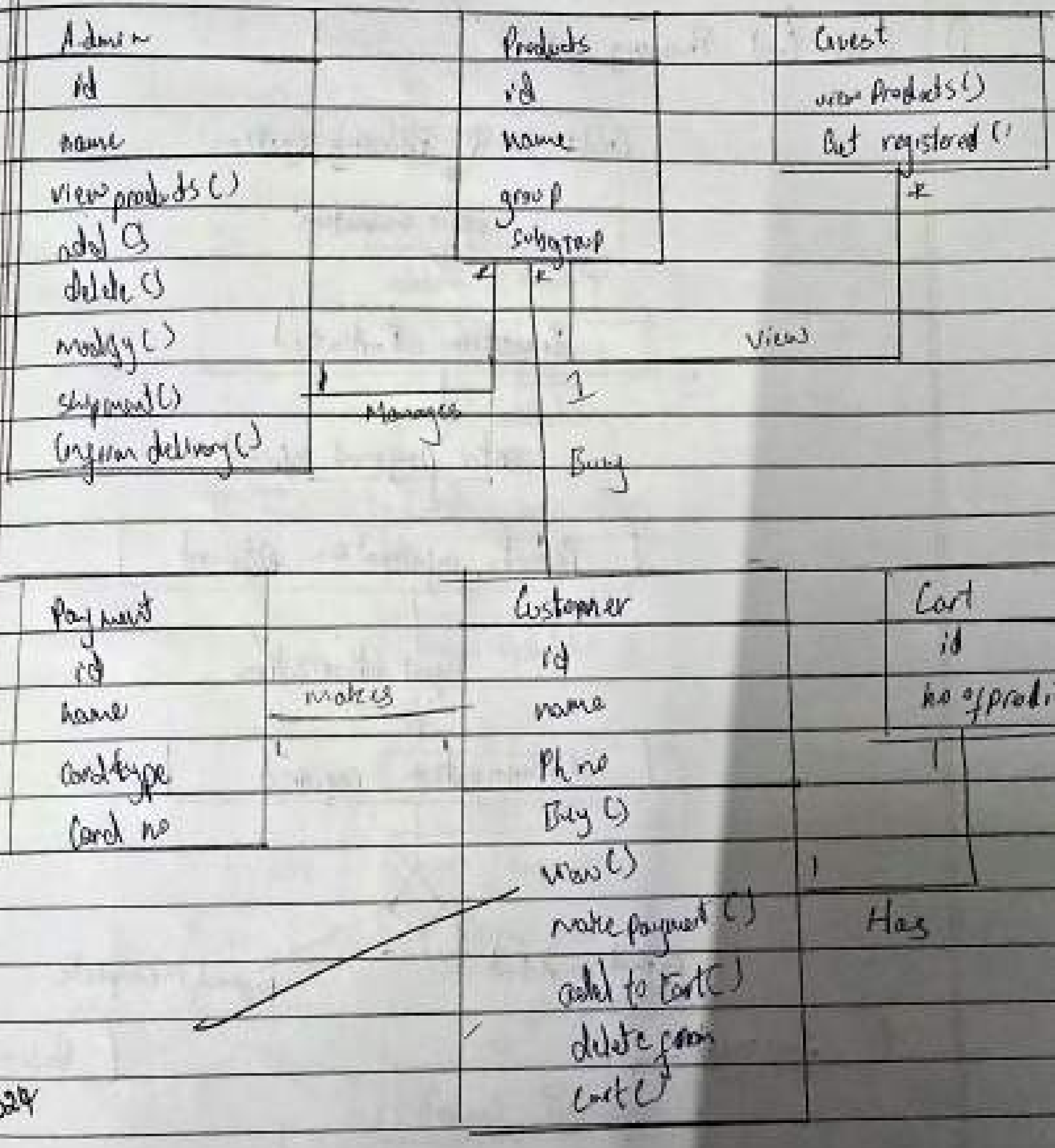
Regular PassPort
fees : float
validity period : int
getfees()
get validity period()

Diplomatic passport
embassy : string
privileges : string
get embassy()
get privilege()

Railway Reservation System



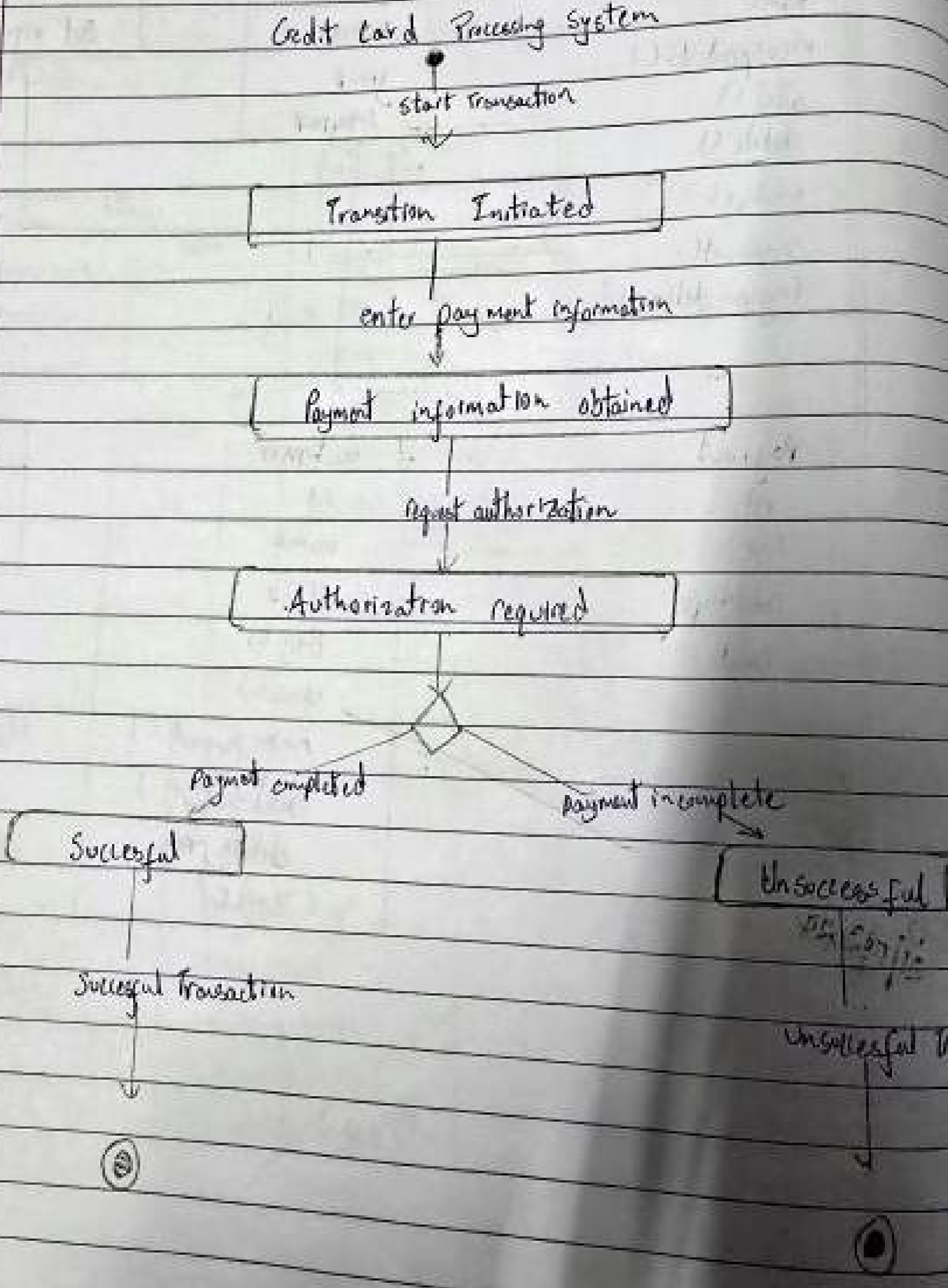
7 Online Shopping System



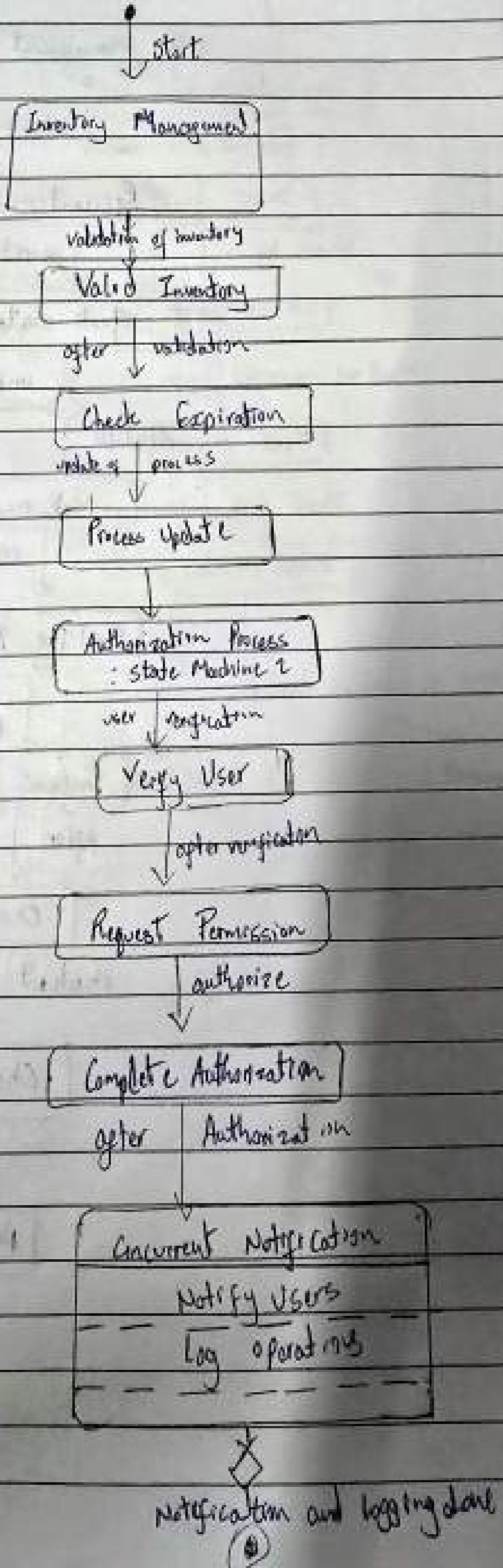
21/10/2024

STATE Machine Diagrams

1) Credit Card Processing system

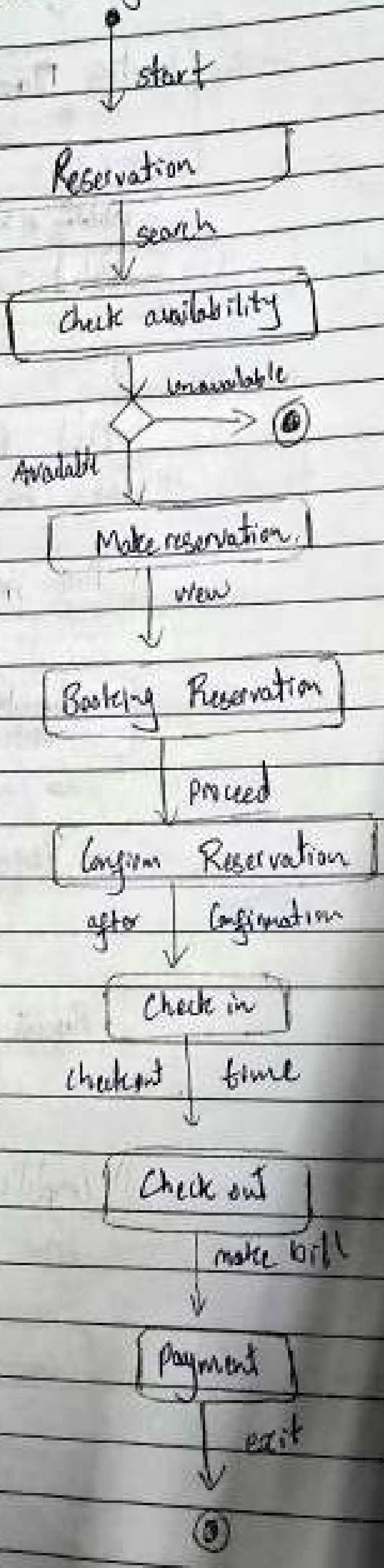


2) Stock Maintainence System

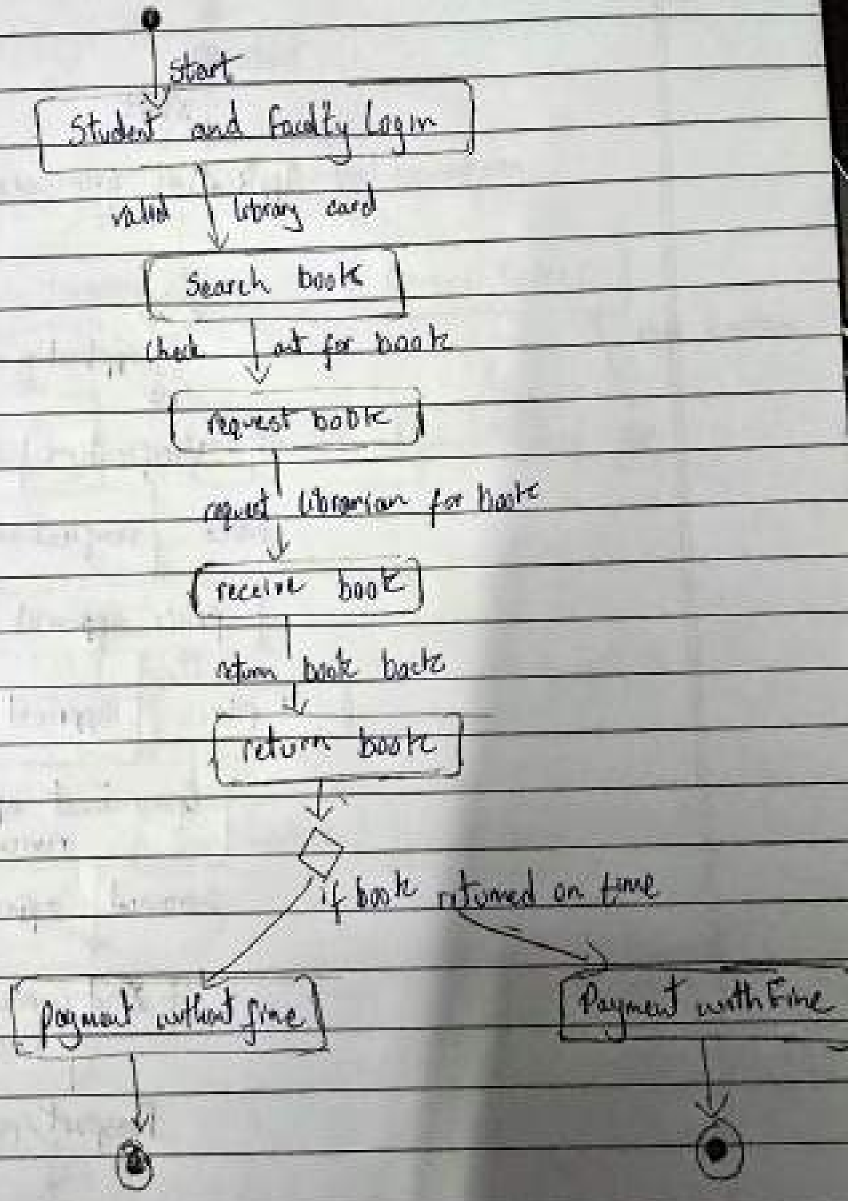


Hotel Reservation System

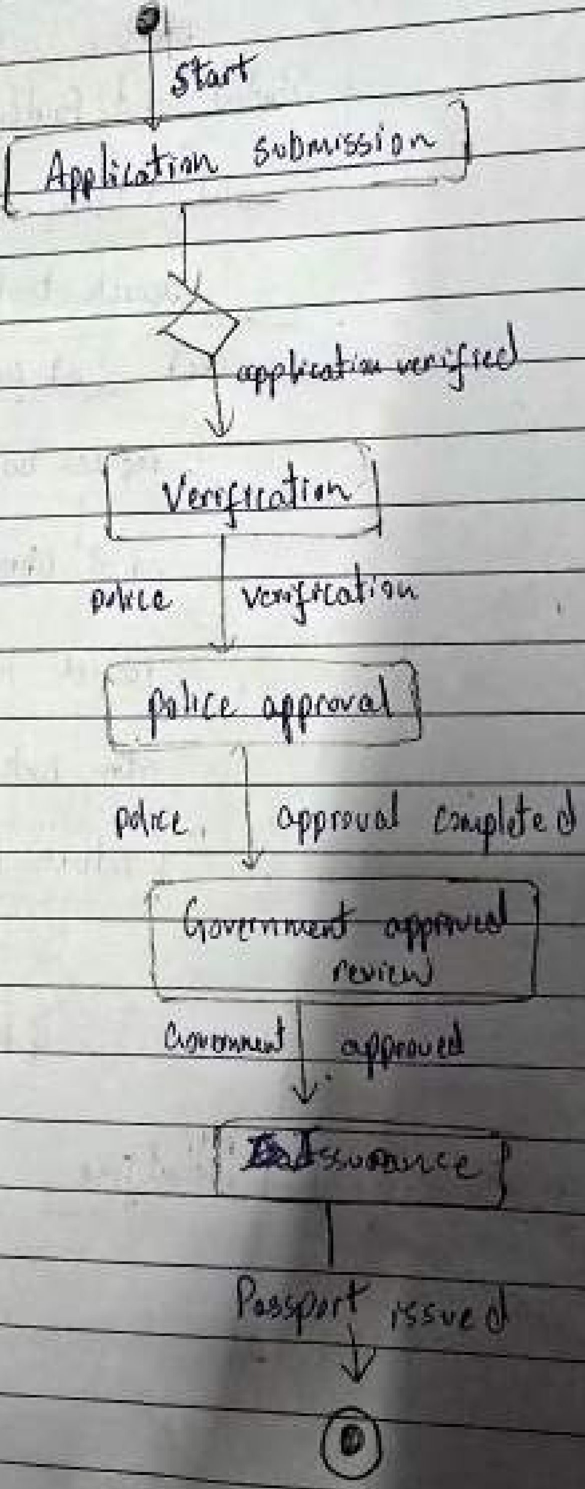
Hotel Management System



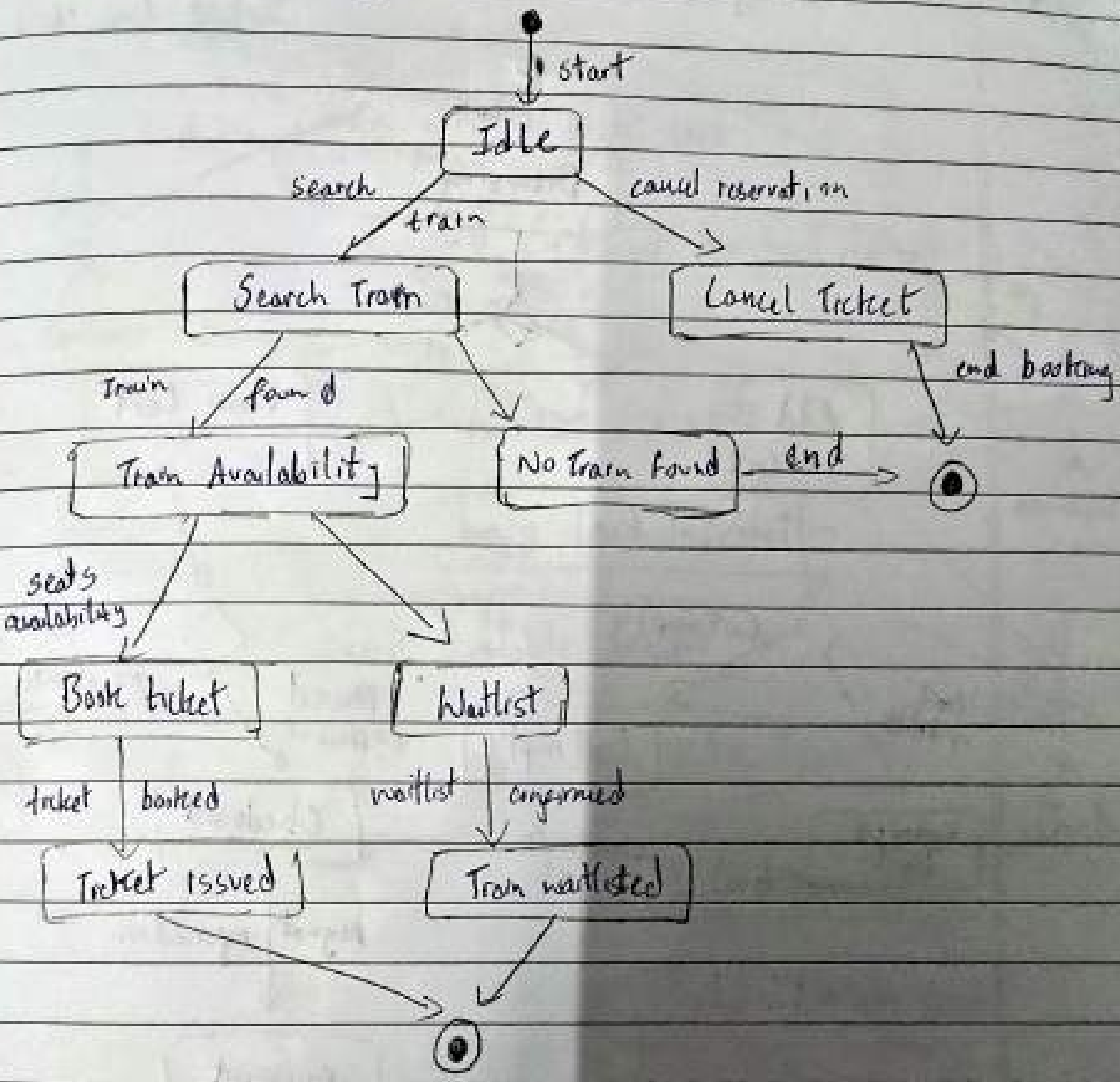
4) Library Management System



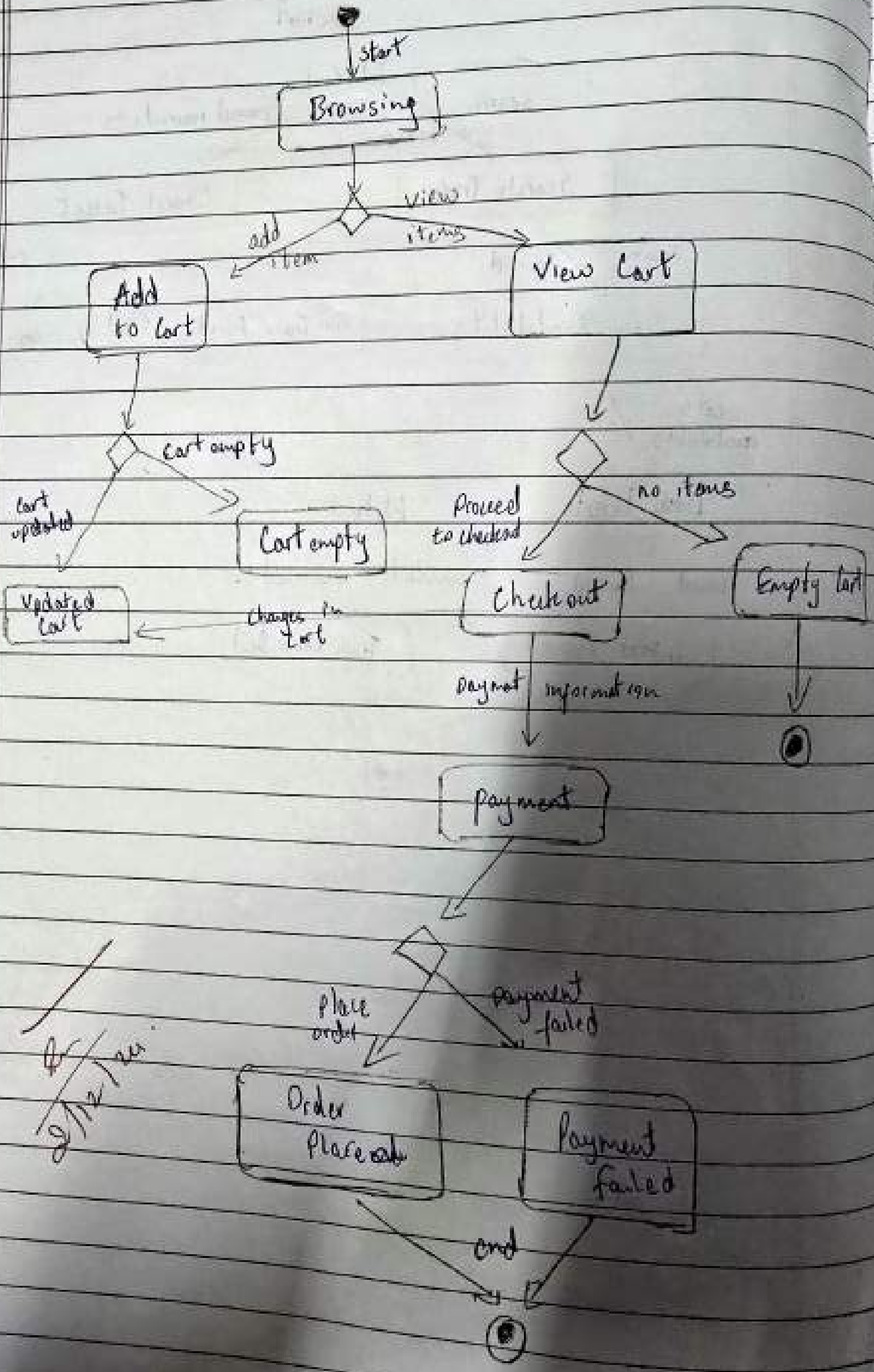
Passport Automation System



6) Railway Reservation System

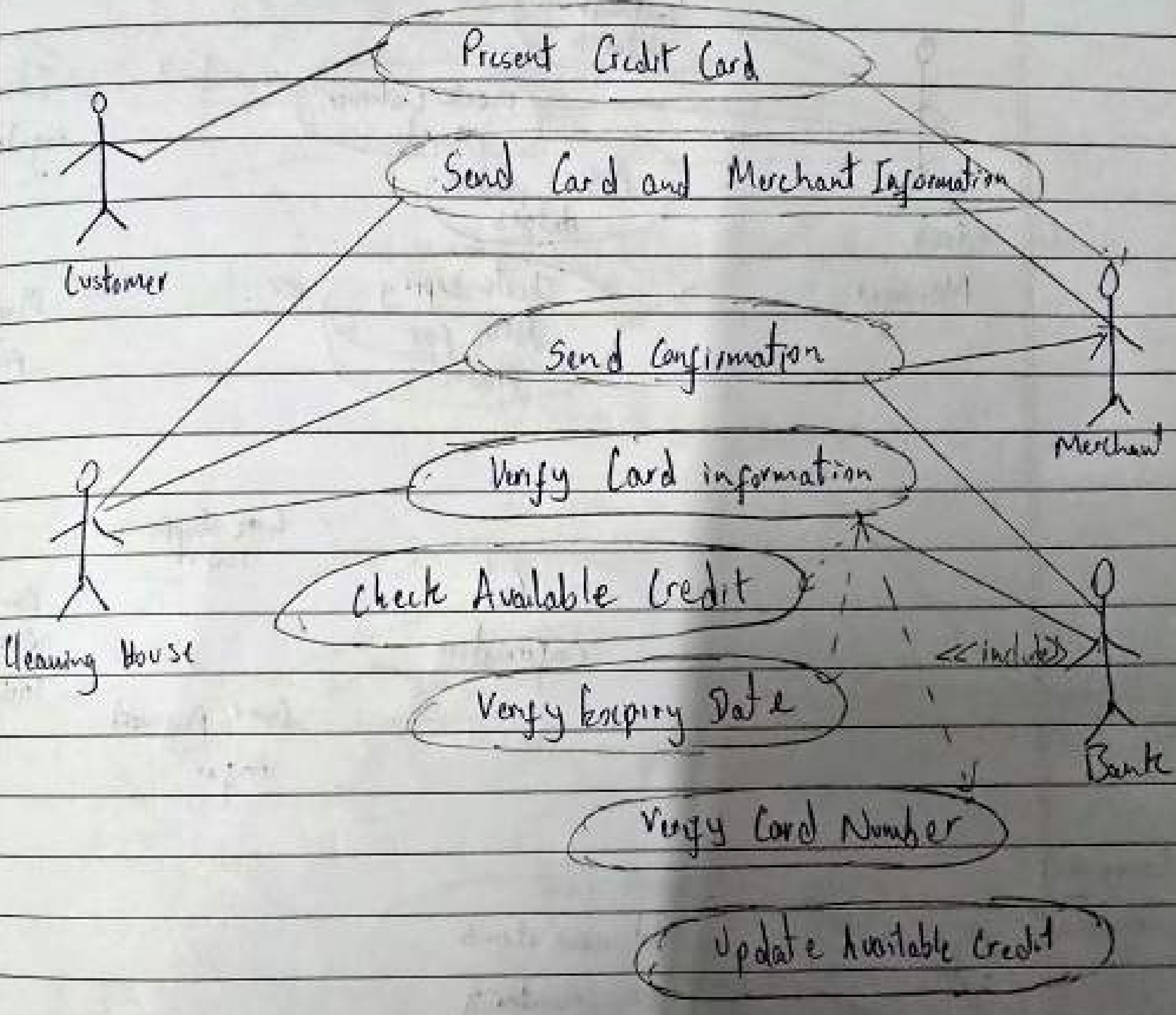


2) Online Shopping System

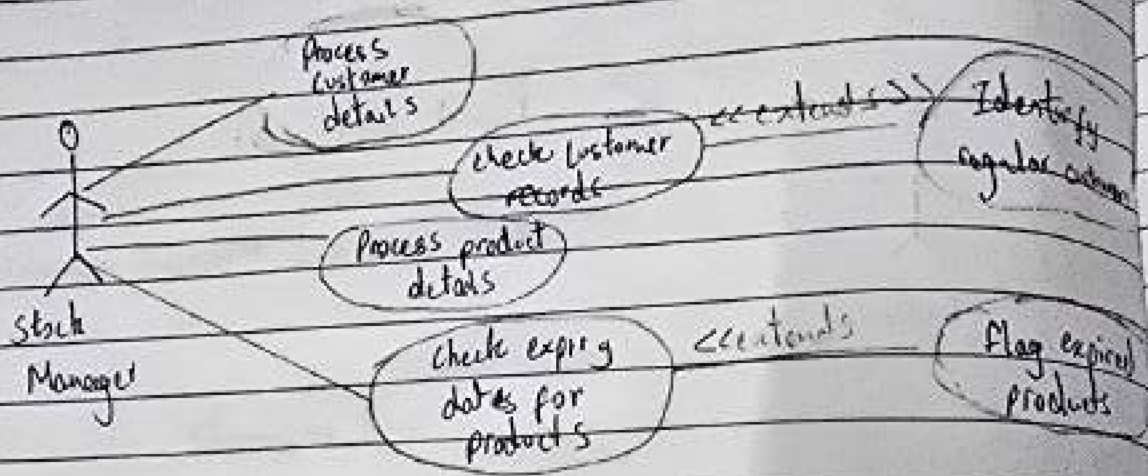


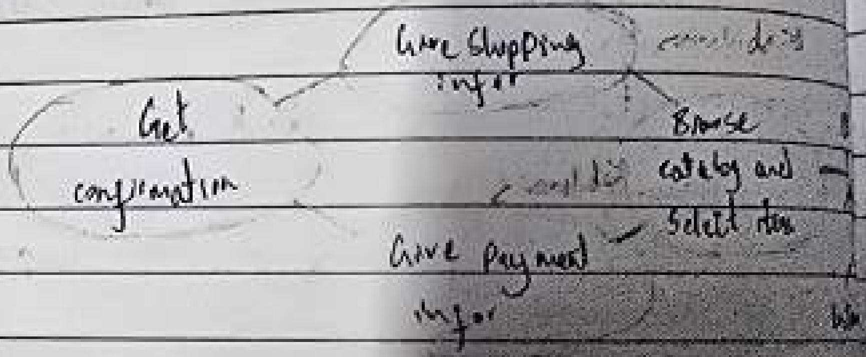
Use Case Diagrams

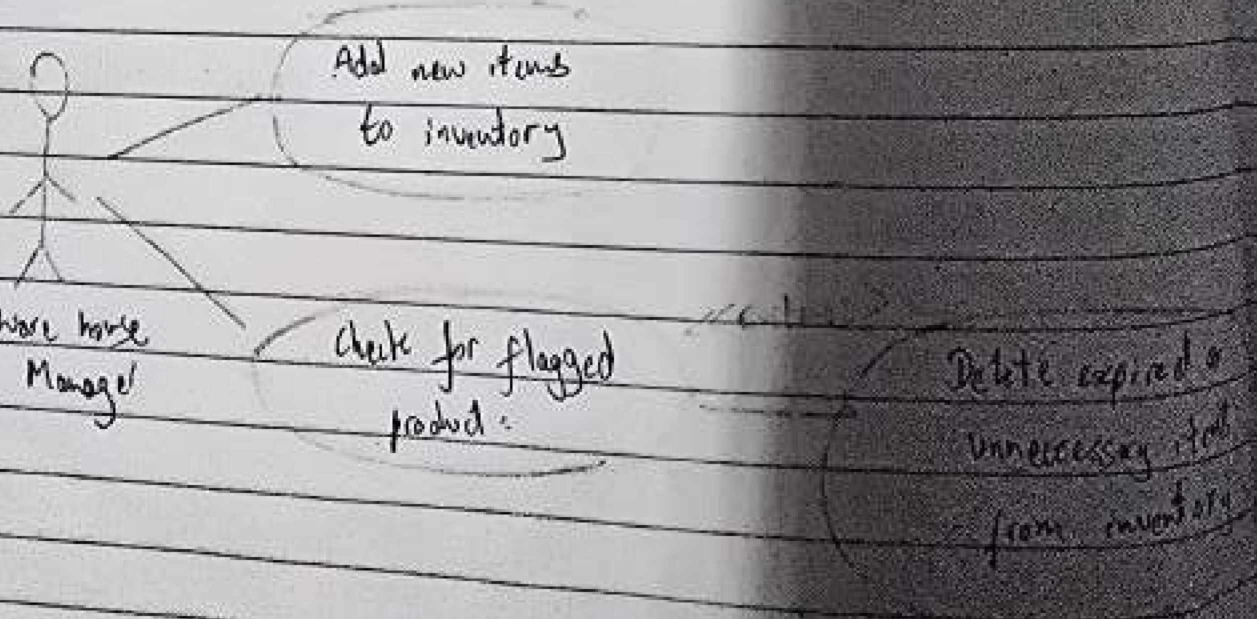
D) Credit Card System



2) Stock Management System

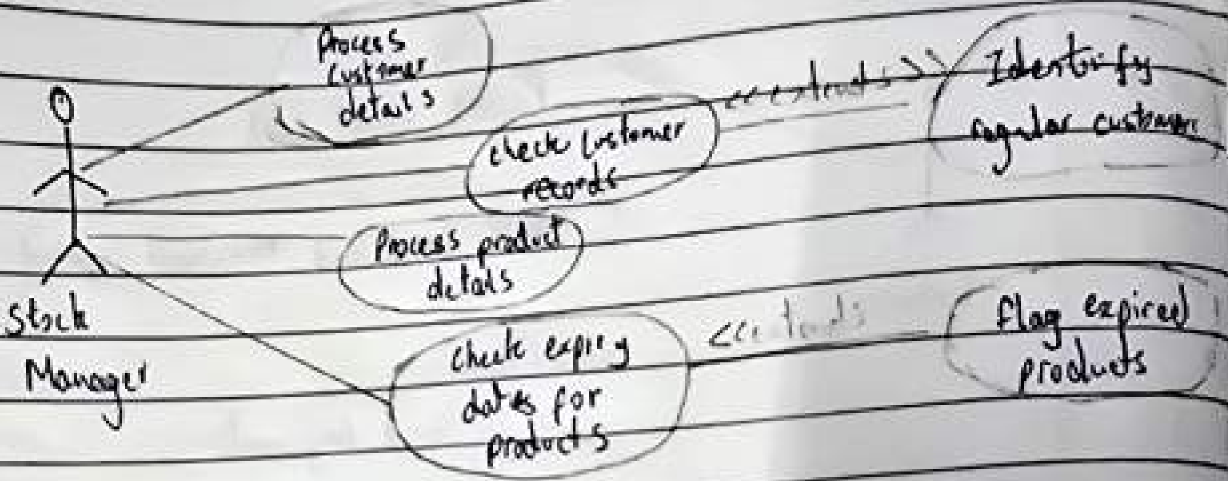


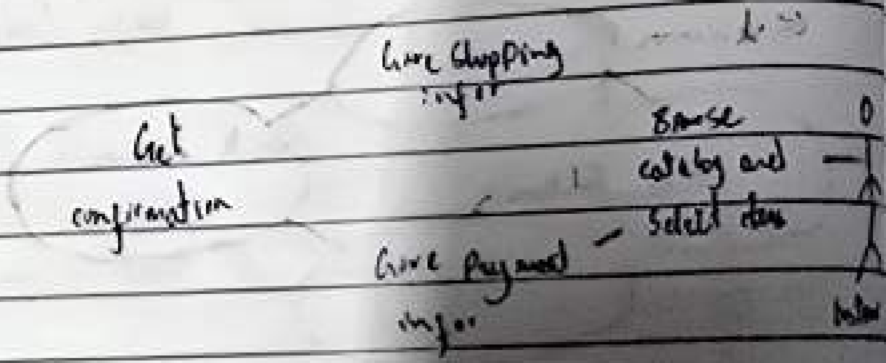


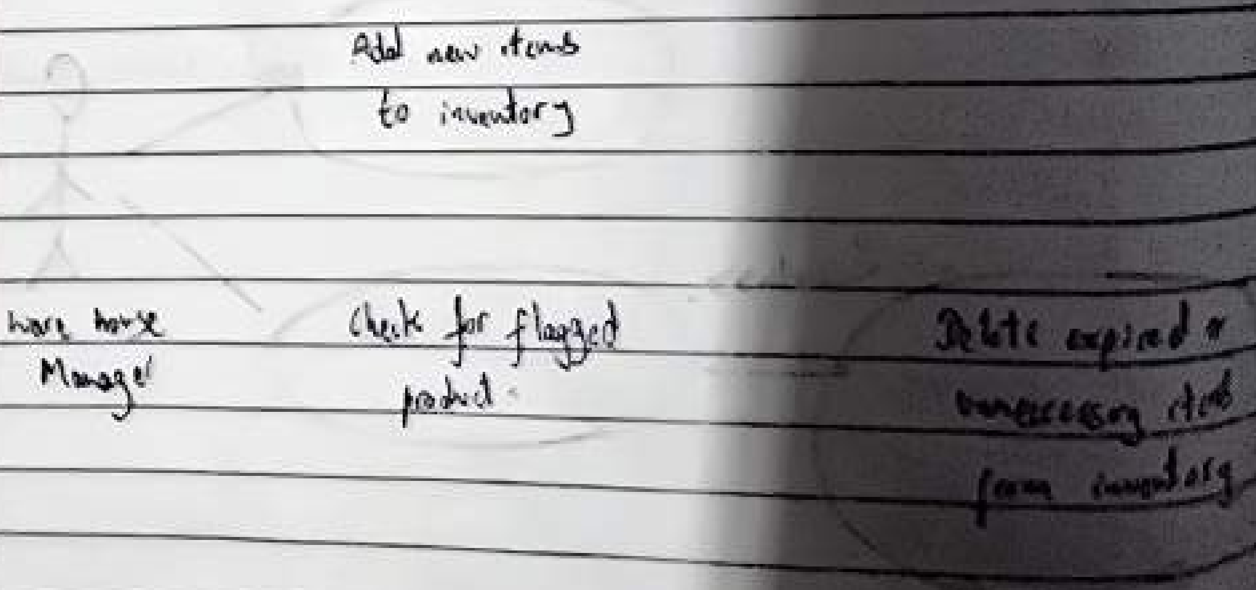

Add new items to inventory
Ware house Manager
Check for flagged product.
<<[illegible]
Delete expired or unnecessary items from inventory

3) Poss

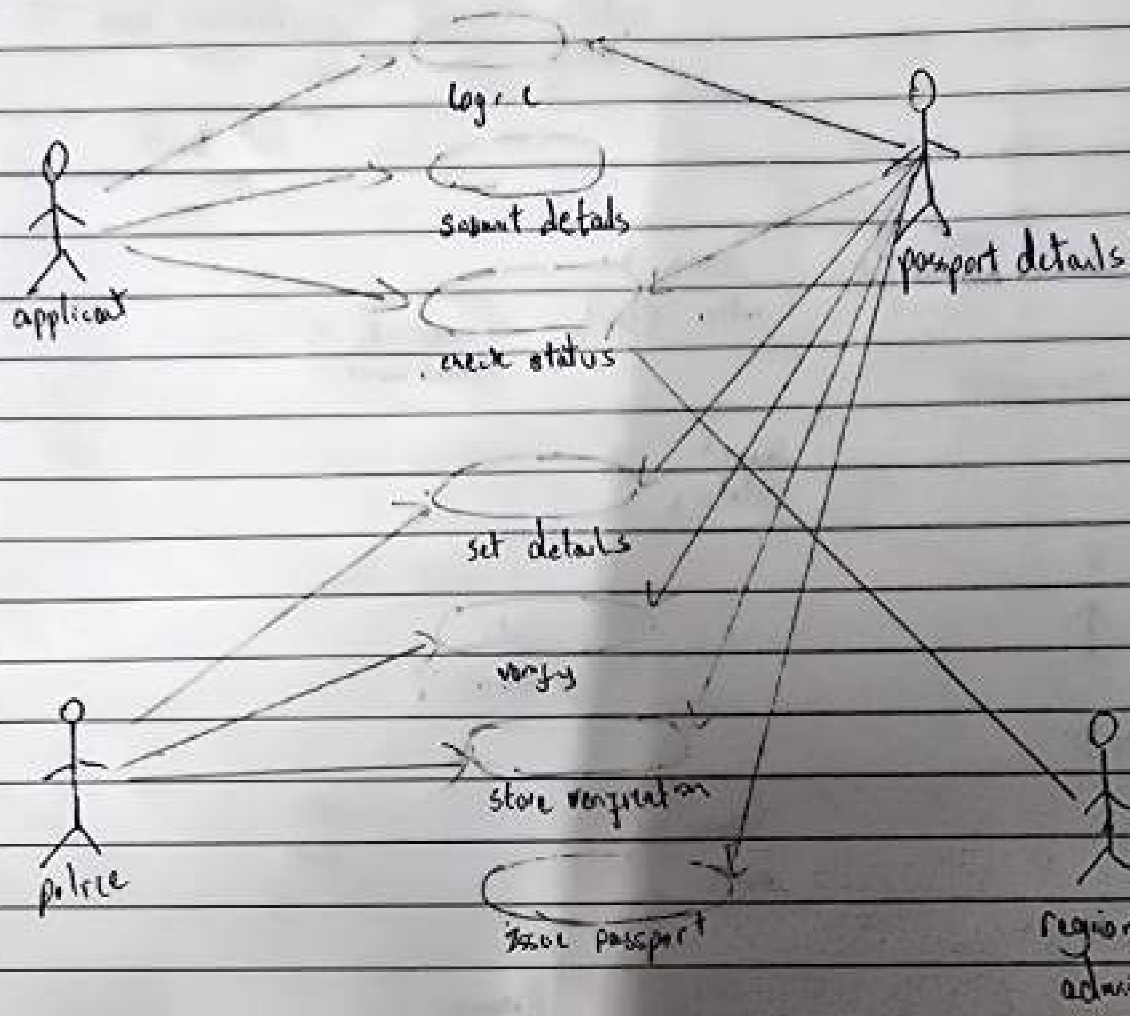
2) Stock Management System



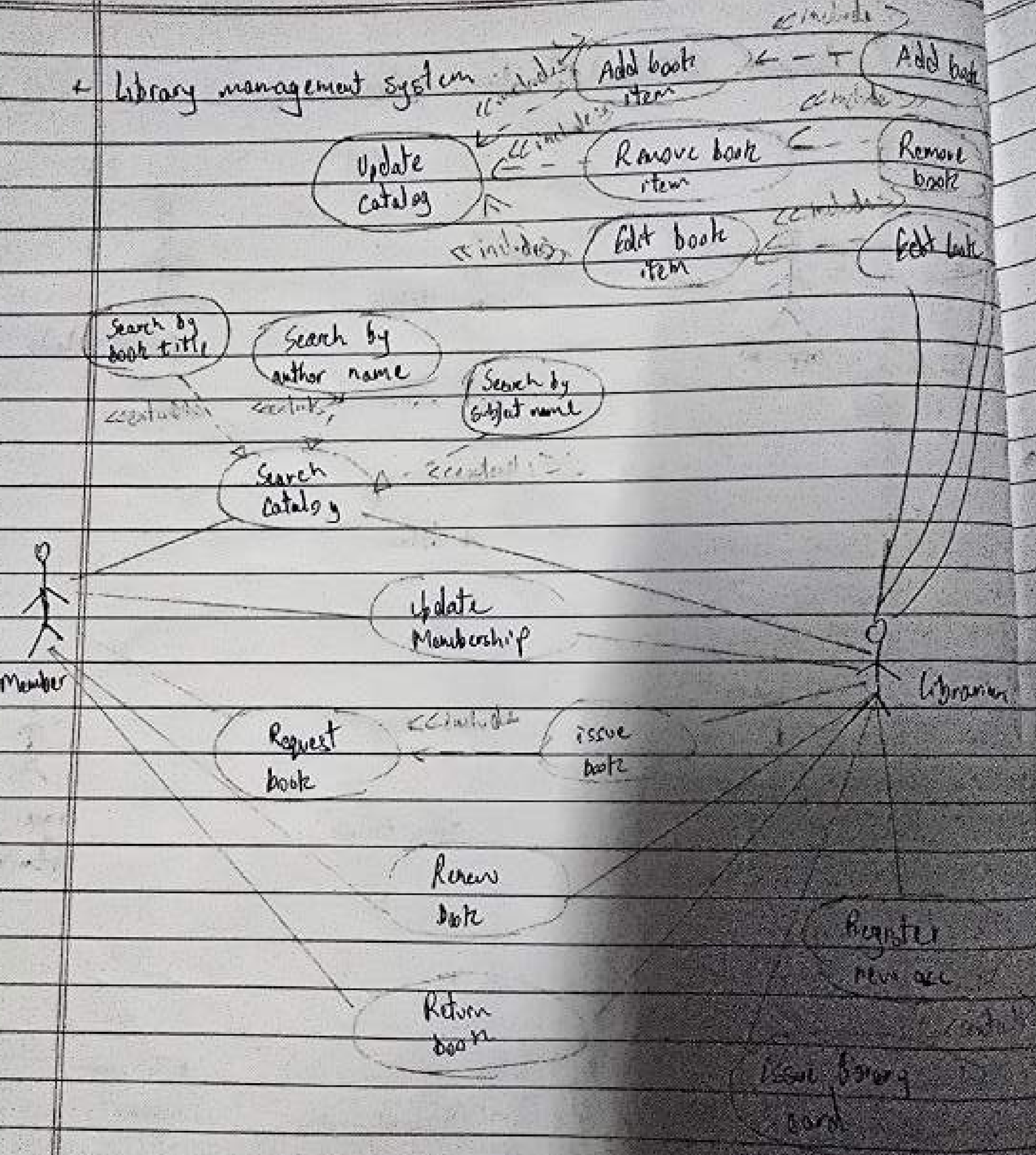



Add new items to inventory
Check for flagged product
Delete expired or unnecessary items from inventory
ware house Manager

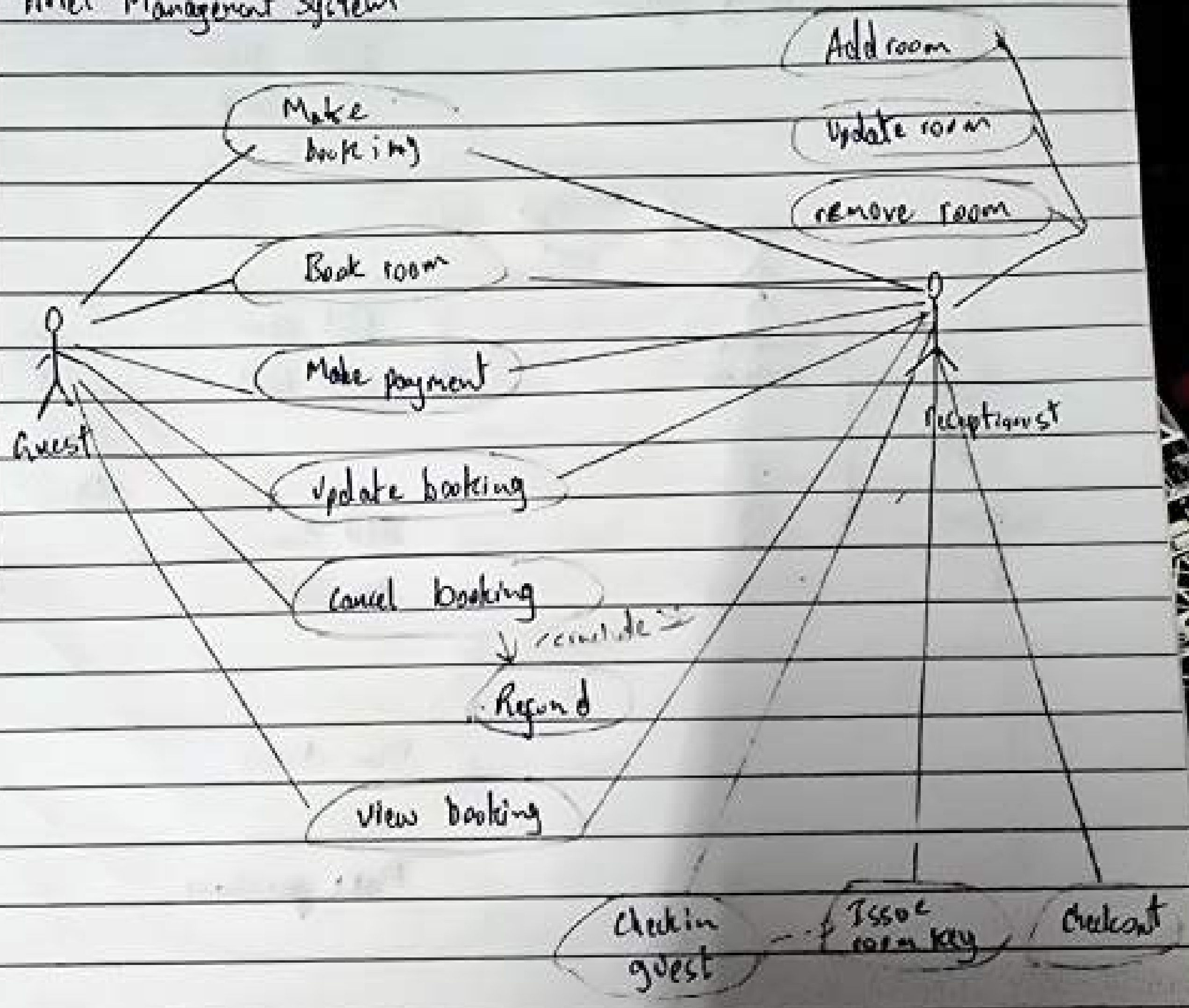
5) Passport Automation System

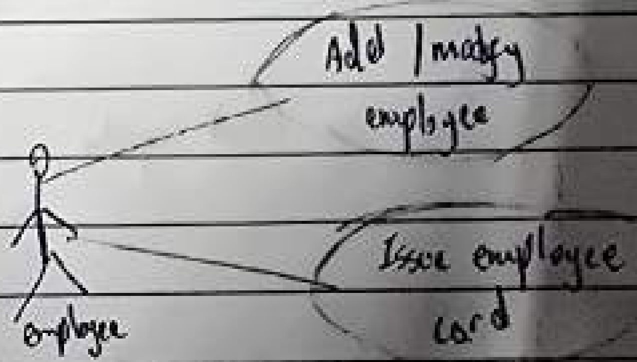


- Library management system

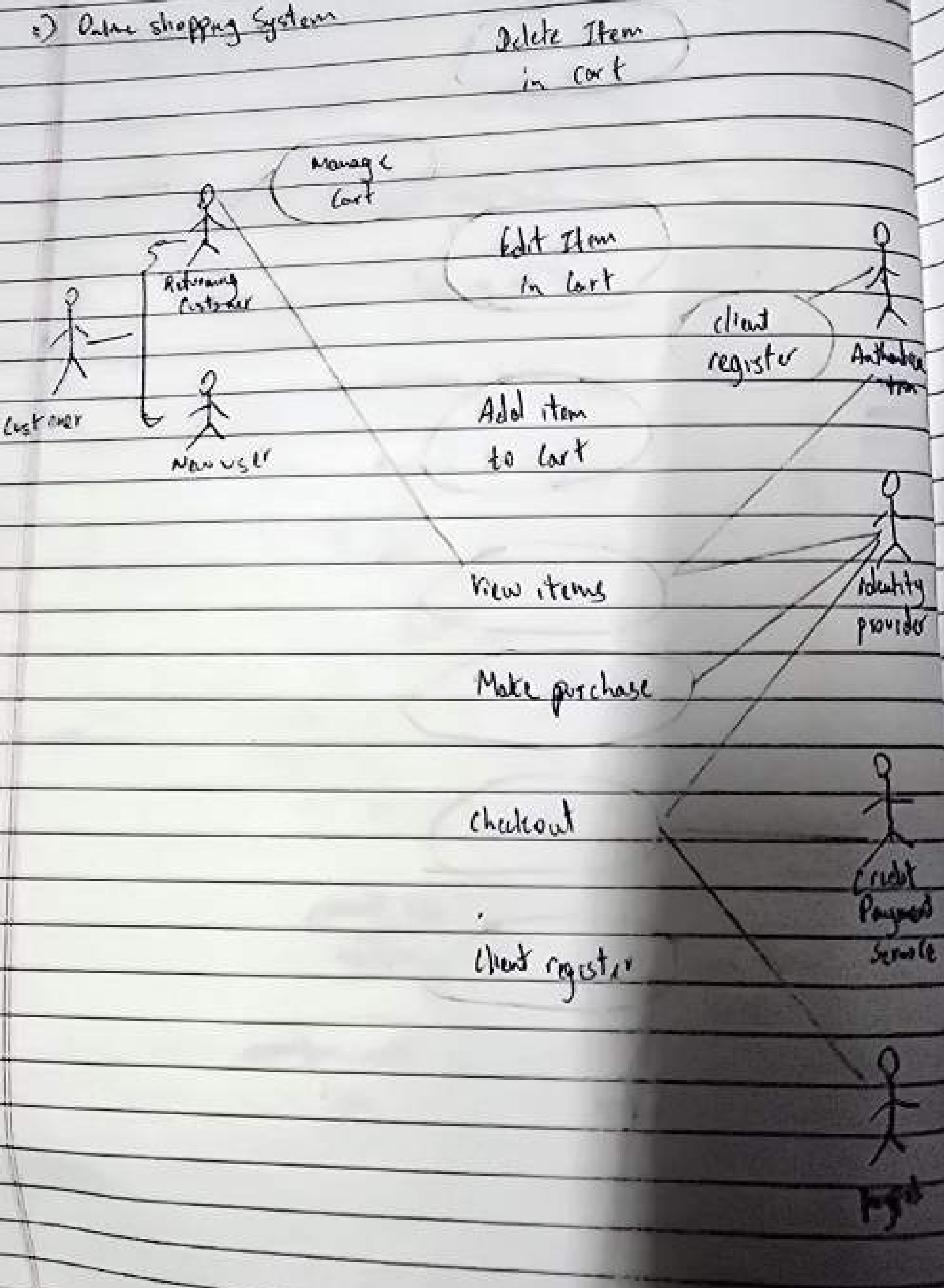


5) Hotel Management System



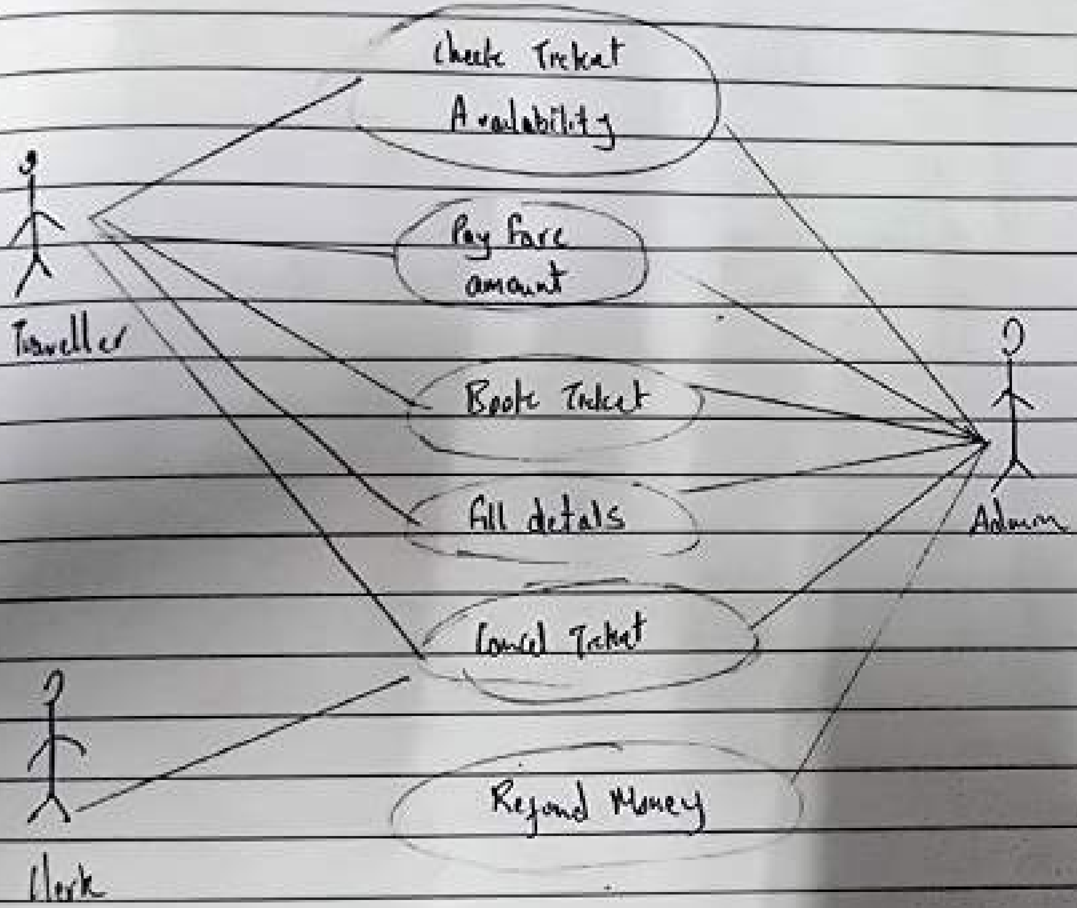


1) Online shopping System



2) Ro

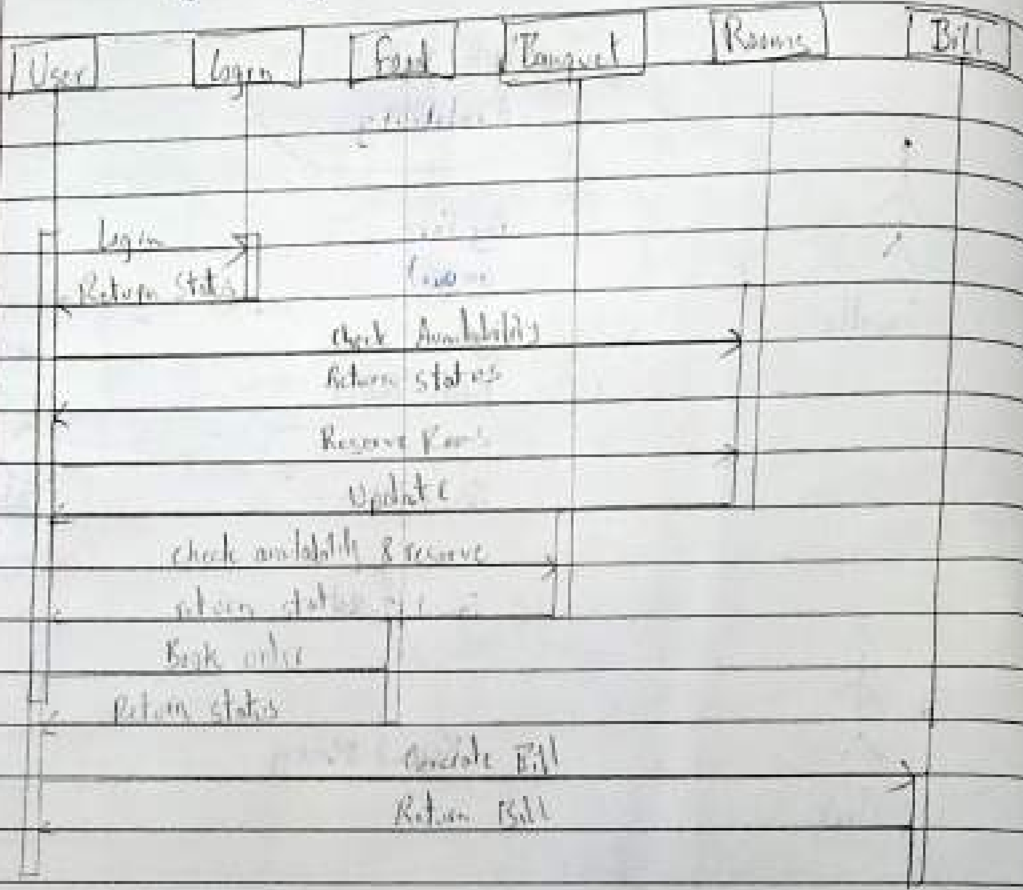
7) Railway Reservation system



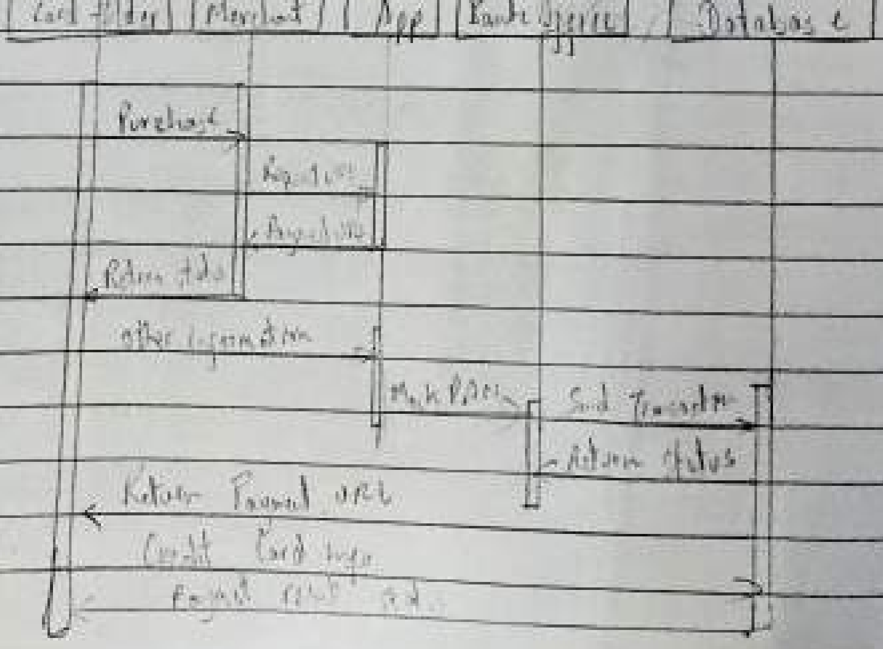
2/12/24.

ix) Sequence Diagrams

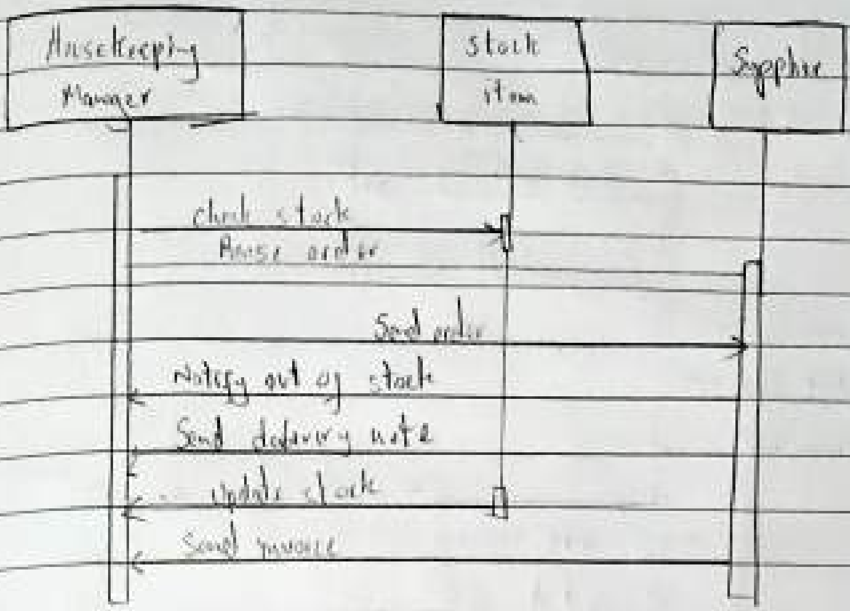
i) Hotel management System



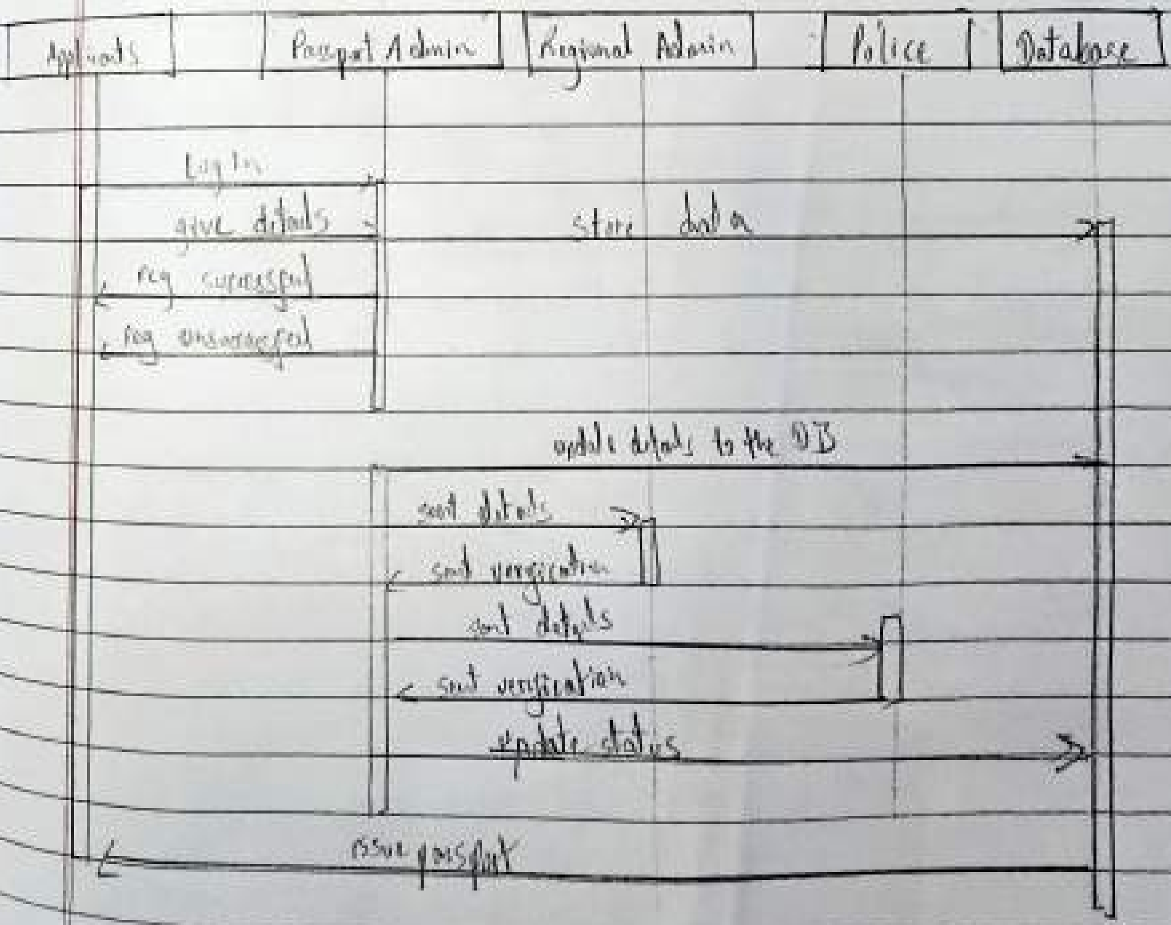
ii) Credit Card Processing System



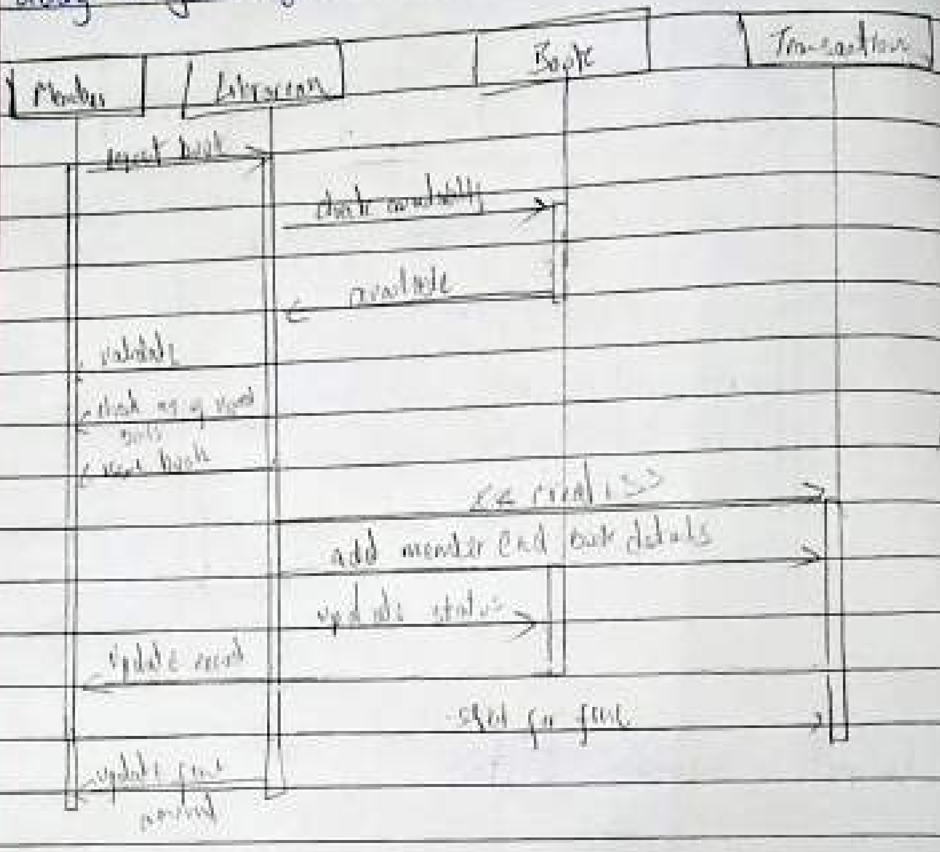
iii) Stock Maintenance System



iv) Passport Automation System

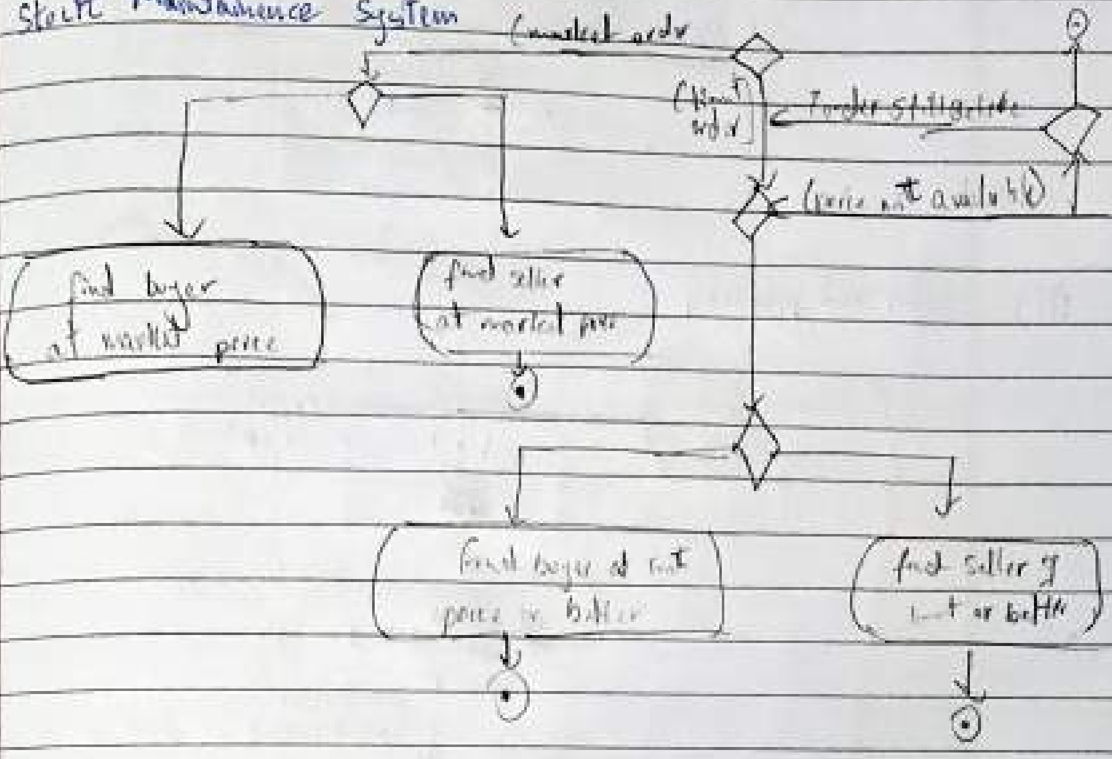


i) Library Management System

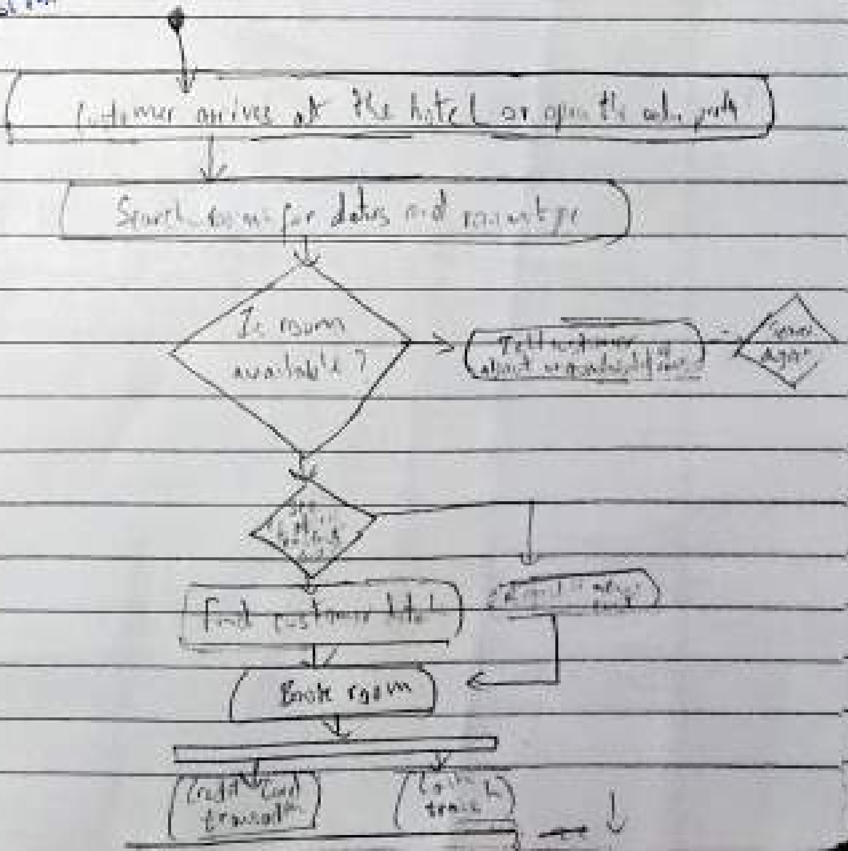


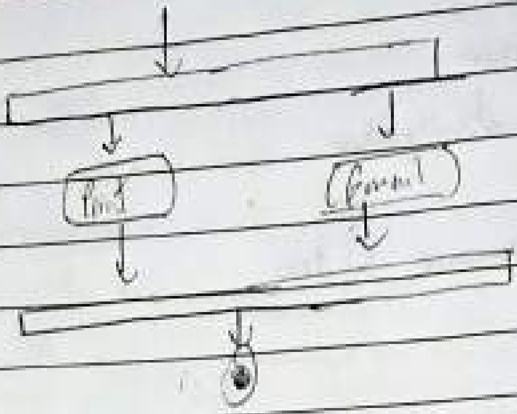
Activity Diagram
[illegible] ~~[illegible]~~ management system

i) Stock Maintenance System

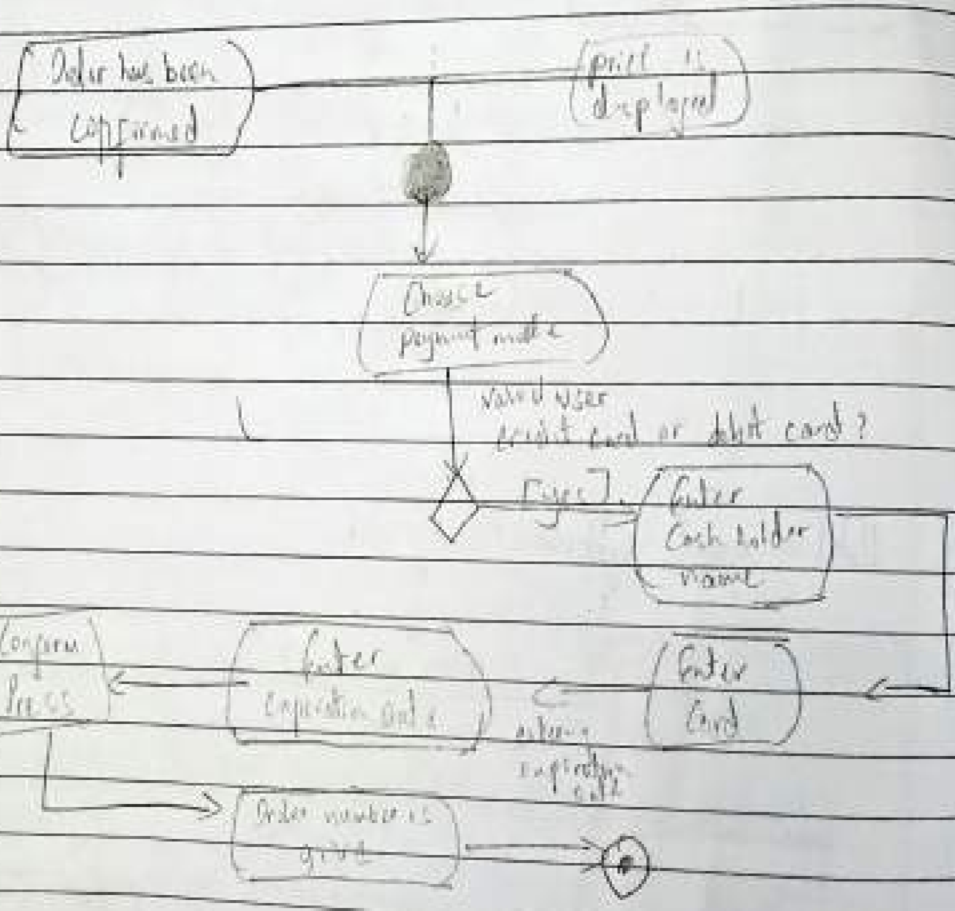

(market order)
(limit order)
Tender still good
(price not available)
find buyer at market price
find seller at market price
find buyer at cost price or better
find seller at cost or better

ii) Hotel management System

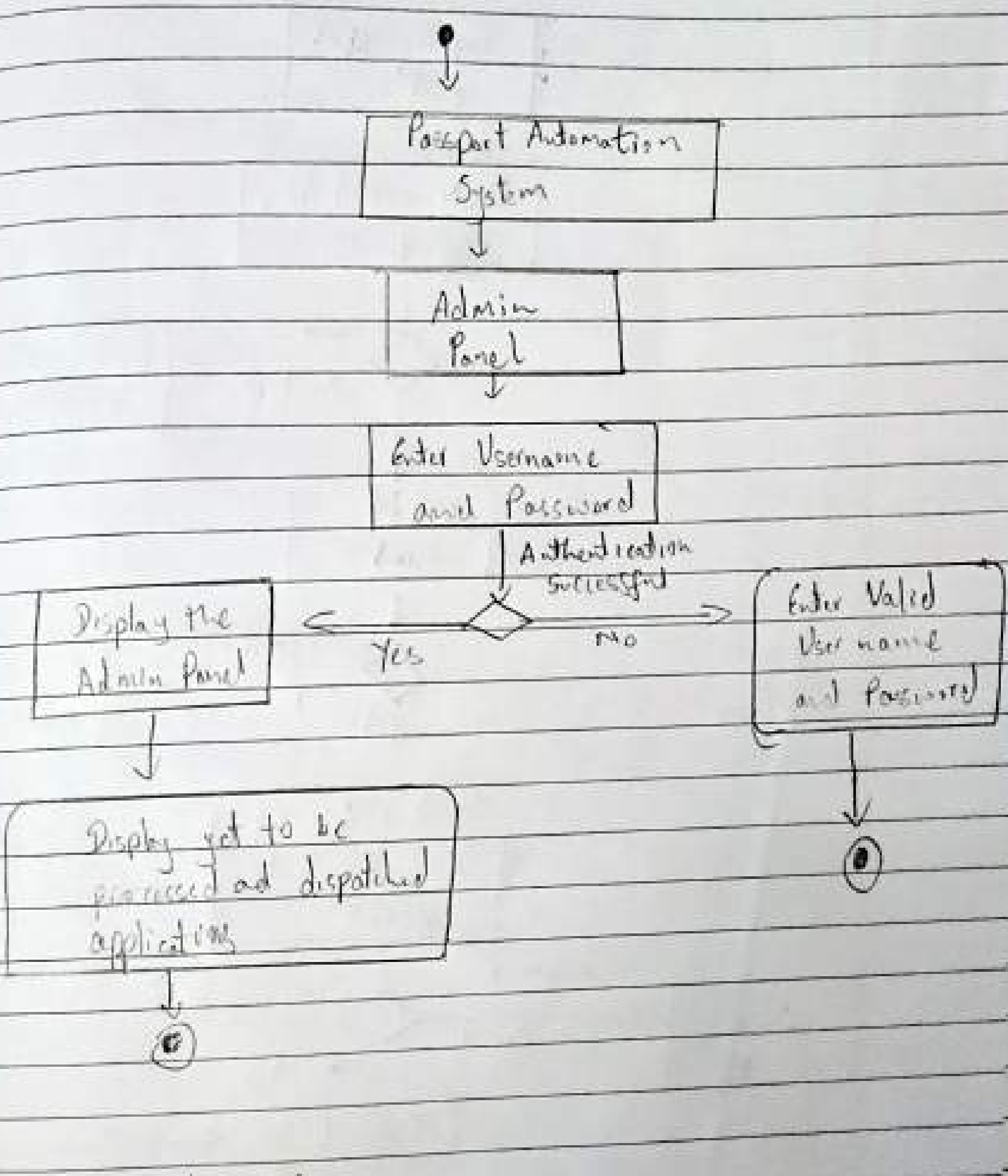

Customer arrives at the hotel or open the web page
Search rooms for dates and room type
Is room available ?
Tell customer about unavailability
[illegible]
Find customer detail
Book room
Credit card transaction
[illegible] transaction



Credit card processing

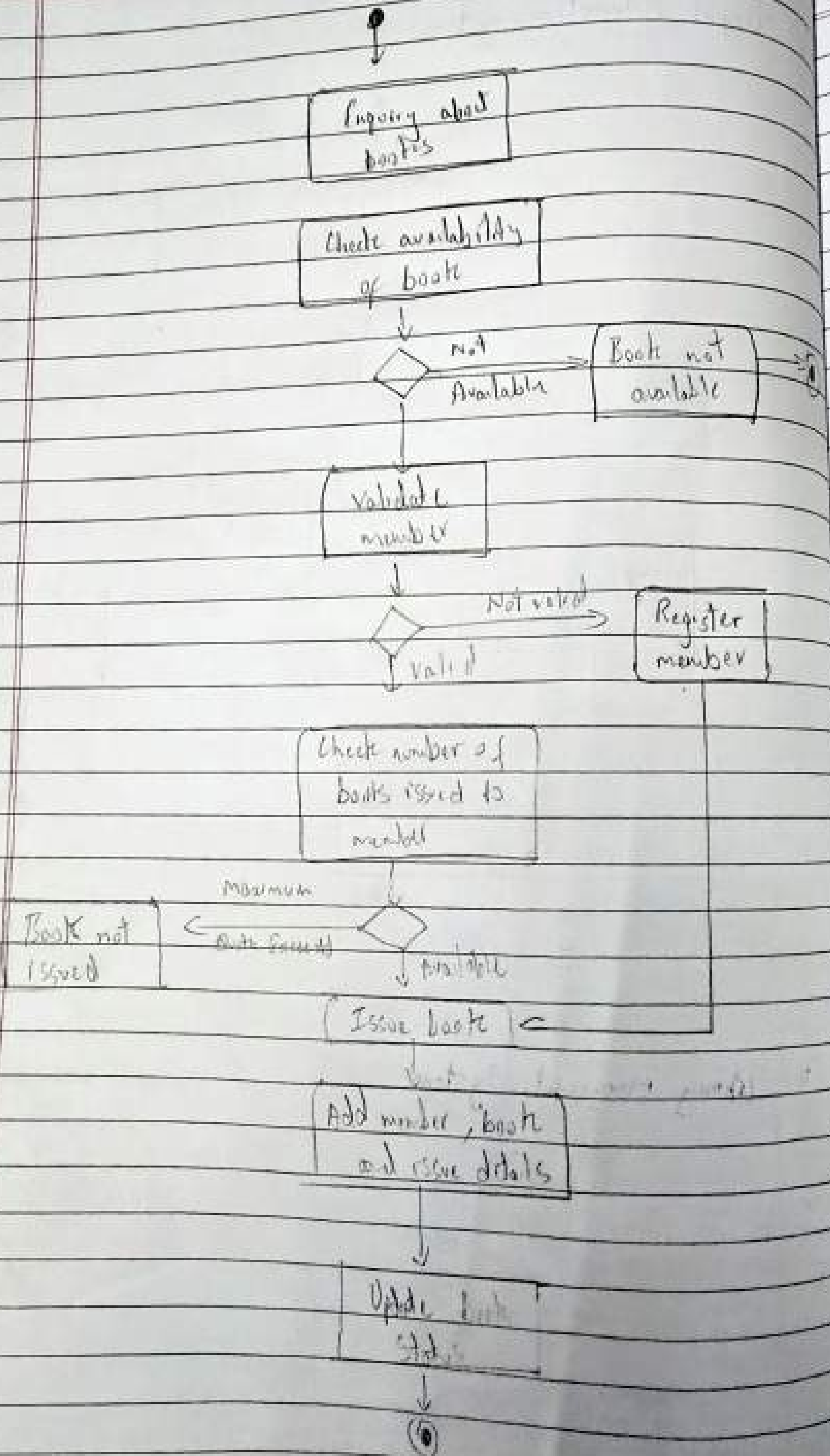


iv) Passport Automation system:-


Passport Automation System
Admin Panel
Enter Username and Password
Authentication successful
Yes
No
Display the Admin Panel
Enter Valid User name and Password
Display yet to be processed and dispatched application

(v) Library Management System

=>


Enquiry about books
Check availability of book
Not Available
Book not available
Validate member
Not valid
Register member
Valid
Check number of books issued to member
Maximum
Quota Issued
Book not issued
Available
Issue book
Add member, book and issue details
Update book status