Date / /
Page

Software Requirement Specification

1. Hotel Management System

1. Introduction

1.1 Purpose: This document specifies the software requirements for a Hotel Management System (HMS). It aims to detail the system's functionality, ensuring smooth and efficient operations for managing bookings, room assignments, billing and housekeeping. The document will serve as a refrence for developers and stakeholders.

1.2 Scope

The HMS will facilitate seamless Hotel operations by enabling staff to handle reservations, room mangment and finacial transactions. The system will automate manual tasks, reducing administrative overhead and improving customer service. The project aims for a 6-month development cycle, with an estimated cost of $100000 offering a reliable solution to hotel managers.

1.3 Overview

This system will offer functionalities such as card authorisation, transaction processing, and reporting. It will interact with external payment processors and provide fraud detection mechanisms.

2. General Description
This is a user friendly platform to manage room bookings, billing, etc offering secure authentication and multi-user support to ensure accurate data handling and overall improve the overall operational efficiency of the system.

3. Functional Requirements
3.1 Reservation Management
This allows to make, modify and cancel reservations with immediate updates on room availability ensuring accurate booking information

3.2 Room Availbility Tracking
It must track real-time room availability for different rates categories (single, double, suite) insuring up-to-date information for staff & customers

3.3 Billing and Invoice
It will automatically generate bills at check-out, including taxes, discounts
- invoice generation
- refunding processing
- payment gateway integration

3.4 Notification and alerts
- Booking confirmation and reminders
- Alerts for room service, maintence, or housekeeping needs.

4. Interface Requirements

- User Interface : interactive UI accessible via web browsers for both hotel staff and customers.

- Database interface: integration with relational database to store information on guests, rooms, transactions and services.

- Payment gateway integration : A secure interface for processing credit/debit card payments.

5. Performance Requirements

↳ System should handle a minimum of 100 concurrent users without performance degradation.

↳ Response time for booking, check-in, and check-out operations must be less than 2 seconds.

↳ System should be able to process up to 500 transactions per day.

6. Design Constraints

↳ System will use a MySQL database

↳ The front end will be developed using web technologies like HTML, Javascript or Python

↳ System must comply with local tax laws and payment processing regulations.

7. Non functional Attributes

- Security: System must ensure security of guest data and payment information using secure access protols.
- Portability: Should be accessible on mobiles and desktops.
- Reliability: System should maintain a high uptime to ensure smooth operations.

8. Preliminary Schedule and Budget

- Development Timeline: 7 months for full system development, testing and deployment.
- Estimated Budget:

CREDIT CARD PROCESSING

1. Introduction

1.1 Purpose

This documents outlines the system overall requirements for the credit card processing system designed to securely process credit cards transactions, manage payments, and ensure fraud detection.

1.2 Scope

The CCPS will handle credit card transactions between customers and merchants. It will ensure secure payments, provide transaction tracking, and integrate with banks for real-time approvals. The system will be optimized for both online and in-store payments.

1.3 Overview

The system enables customers to make payments using credit cards, while merchants can process these transactions securely. It will include features like payment authorization, fraud detection, transaction history, and refund processing.

2. General description

The CCPS will support secure and fast payment processing between merchants and customers. It will handle authorizat settlements of payments, and provide transactions logs for both parties.

3 Functional Requirements

3.1 User Authentication : secure login for customers and merchants.

3.2 Payment Authorisation : Verify credit card details, check balance, and authorize transactions in real-time.

3.3 Transaction Logging : Log transaction details including ID, amount & status.

3.4 Refunds and Chargebacks : Enable merchants to process refunds and customers to dispute transactions.

3.5 Fraud detection : Detect, flag and block suspicious transactions

3.6 Notifications : Send SMS / email for approvals, declines, refunds and chargebacks.

4 Interface Requirements

User Interface (UI) : A secure web or mobile interface for customers and merchants to process and view transactions.

Bank API integration : Seamless integration with banks for real-time transaction authorizations.

Encryption : Secure all transactions with SSL/TLS encryption.

Date __/__/__
Page _____

5. Performance Requirements :

- The system must process transactions within 2 seconds.
- It should support up to 10,000 concurrent users during peak hours.
- Handle 1 million transactions daily.

6. Design constraints

- The system must comply with PCI DSS (Payment Card Industry Data Security Standard) to ensure data security and privacy
- Use secure coding practices to prevent data breaches and ensure high availability.

7. Non-functional Attributes :

- Security : End-to-end encryption for all sensitive data, including credit card numbers and personal information.
- Reliability : 99.99% uptime to ensure consistent payment processing.
- Scalability : Ability to handle increasing transactions volumes without performance degradation.
- Compliance : Must comply with local and international financial regulations.

Date __/__/__
Page ____

Preliminary Schedule and Budget

Timeline : 6 months per development, testing, and deployment.

Budget :

Date __/__/__
Page _____

WEEK-2

3) LIBRARY MANAGEMENT SYSTEM

1. Introduction

1.1 Purpose

This document outlines the software requirements for the library Management system, detailing features, functions, and design constraints to ensure the system meets its intended use and delivers value to users and administrators.

1.2 Scope

The LMS will manage library operations like book cataloging, issuance, returns, and user records. It is designed for academic and public libraries to streamline operations, reduce manual efforts, and improve accessibilities. This document also covers development timelines and cost estimates.

1.3 Overview

The LMS will digitize library operations, including book searches, reservations, and overdue alerts. It aims to simplify librarian tasks and enhance user experience.

2. General Description

- User Characteristics: students, faculty, and public
- Features: Book search, track issuance/returns, genrate reports.
- Key Benefits: Reduces librarian workload and ensures efficient resource tracking.
- Importance: Enhances user satisfaction and streamlines library management.

Date __/__/__
Page _____

3. Functional Requirements

- Book Magement : Add, update, or remove books. The librarian will be able to enter details like ISBN, author, title, etc.

- User management
Facilitate registration of new users and manage their profiles. Each user will have a unique ID and their activities can be tracked using it. Additionally the system will also allow admins to revoke privileges.

- Issuance & returns
The system will track the issuance and returns of the books. It will alert the users when books are overdue and calculates fine if needed. Upon returning the book it calculates the total cost and updates the necessary details in the database.

- Search functionality
The LMS will provide a robust search function to easily locate books and browse collections based on their interests.

- Report generation
Generates reports on issued books, overdue books and fines. Also tracks library inventory and book circulation.

Date __/__/__
Page

4. Interface requirements
- User interface (UI): A web-based or desktop application with user friendly navigation for both librarians and users.
- Database interface: integration with a relational database or to store information on books, users and transactions.
- Notification system: email or SMS integration to send reminders for due dates, overdue books and fines.

5. Performance requirements
- system should support up to 200 concurrent users without performance issues
- issuing and returning books should be processed under 5 seconds.
- system should handle up to 1000 transactions per day.

6. Design constraints.
- system must comply with library data ~~mss~~ management standards.
- It will be developed using a relational database and a web-based frontend.
- system should be portable across different devices.

7. Non-functional Attributes
- Security: The system should use encryption to protect user data and enforce secure login credentials.
- Reliability: system should have an uptime of 99%. to ensure smooth functioning.
- Scalability: scalable to accomodate future increases.
- portability: should be compatible with various web browsers.

Date __/__/__
Page _____

) Preliminary Budget and Schedule

i. Timeline: 4 months for full development and 1 month for testing.

Budget: ₹ 1,000,0000 which includes development and testing costs

Date __/__/__
Page

4) STOCK MAINTAINANCE

1) Introduction

1.1 Purpose of this document

This document will outline the systems functional and non functional requirements and provide a detailed description for efficient management.

1.2 Scope of this document

The main objective is to track inventory levels, generate alerts, provide reports, etc. Thus it aims to reduce manual efforts which will minimize errors.

1.3 Overview

This will allow businesses to track inventory in real-time, manage supplier details, record stock entries, etc.

2) General description

This a desktop app for real time inventory tracking, stock ~~dt~~ alerts, and reporting. It helps users manage stock more efficientey and reduce manual errors.

3) Functional Requirements

↳ a) Stock tracking
- add, edit, delete and categorize stock items by various fields.

↳ b) Stock alerts : notification sent when stock falls below threshold.

↳ c) Stock entry / exit : record stock entries and exits with date, time, and quantity details.

↳ d) Supplier management : allows users to manage supplier information.

Date __/__/__
Page

4) Interface Requirements

↳ User Interface:
The system should have a graphical user interface with forms of stock management.

↳ interface should allow users to navigate between stock lists, reports and supplier management.

↳ Software interface: should integrate with the existing accounting system via an API.

5) Performance Requirements
- should be able to handle up to 10,000 stock items without any performance degradation.
- stock transactions should be processed within 3 seconds.
- system should generate quick reports.

6) Design Constraints
- The system must be developed using .NET framework
- The system should support integration with the existing accounting software.
- The system will be hosted on a cloud server.

7) Non-Functional Attributes
- Security: All stock data shall be encrypted in transit and at rest
- Reliability: The system shall be operational 24/7, with a maximum downtime of 1 hour per month.
- Scalability: minimal performance impact
- Portability: The system shall be accessible from both desktop and mobile devices.

Date __/__/__
Page _____

8) Preliminary Schedule and Budget
- Development Timeline : 4 months.
- Estimated cost: ₹16,00,000 includes development, testing and deployement.

PASSPORT AUTOMATION SYSTEM

1. Introduction

1.1 Purpose of this document

This document outlines the requirements for developing a Passport Automation System (PAS) aimed at streamlining the process of applying for, tracking, and issuing passports.

1.2 Scope of this document

The system will enable users to apply online, schedule appointments, track application status, and receive notifications. It will facilitate document verifications, approval, etc. The goal is to reduce processing time and improve service accuracy.

1.3 Overview

It will handle tasks like application submissions, document verification, biometric capturing and passport dispatch.

2. General description

The system will automate the process of passport application, verification and issuance. Overall it will help in reducing manual power and facilitate quicker tasks.

Date / /
Page

3. Functional Requirements

↳ User Registration and Profile : register new users and update existing user and scheduling appointment for document verification or biometric submission.

↳ passport application : submit applications online with required documents and biometric.

↳ Payment processing : pay application fees online and receive a receipt.

↳ Application Tracking : Track status of application with timely notifications.

4. Interface Requirements

↳ User interface (UI) : A web based interface where the user can track the passport application status and for officials to manage applications

↳ Database interface : Integration with a centralised database to store user details, applications, document and transaction data.

5. Performance Requirements

↳ Response time for form submission and status tracking should be under 3 seconds.

↳ Ability to process up to 10,000 passport applications daily.

↳ system should handle upto 5000 concurrent users during peak times.

Date __/__/__
Page _____

Design Constraints

System must comply with government security and privacy regulations regarding personal data.
System should be developed using secure coding standards to prevent data breaches.
integration with national identity systems for document verification.

Non-functional Attributes

Security : The system must ensure encryption of sensitive user data, including biometric and payment info.
Reliability :
Scalability : should be scalable to include future increases.
Portability : System must be accessible via both desktop and mobile devices.

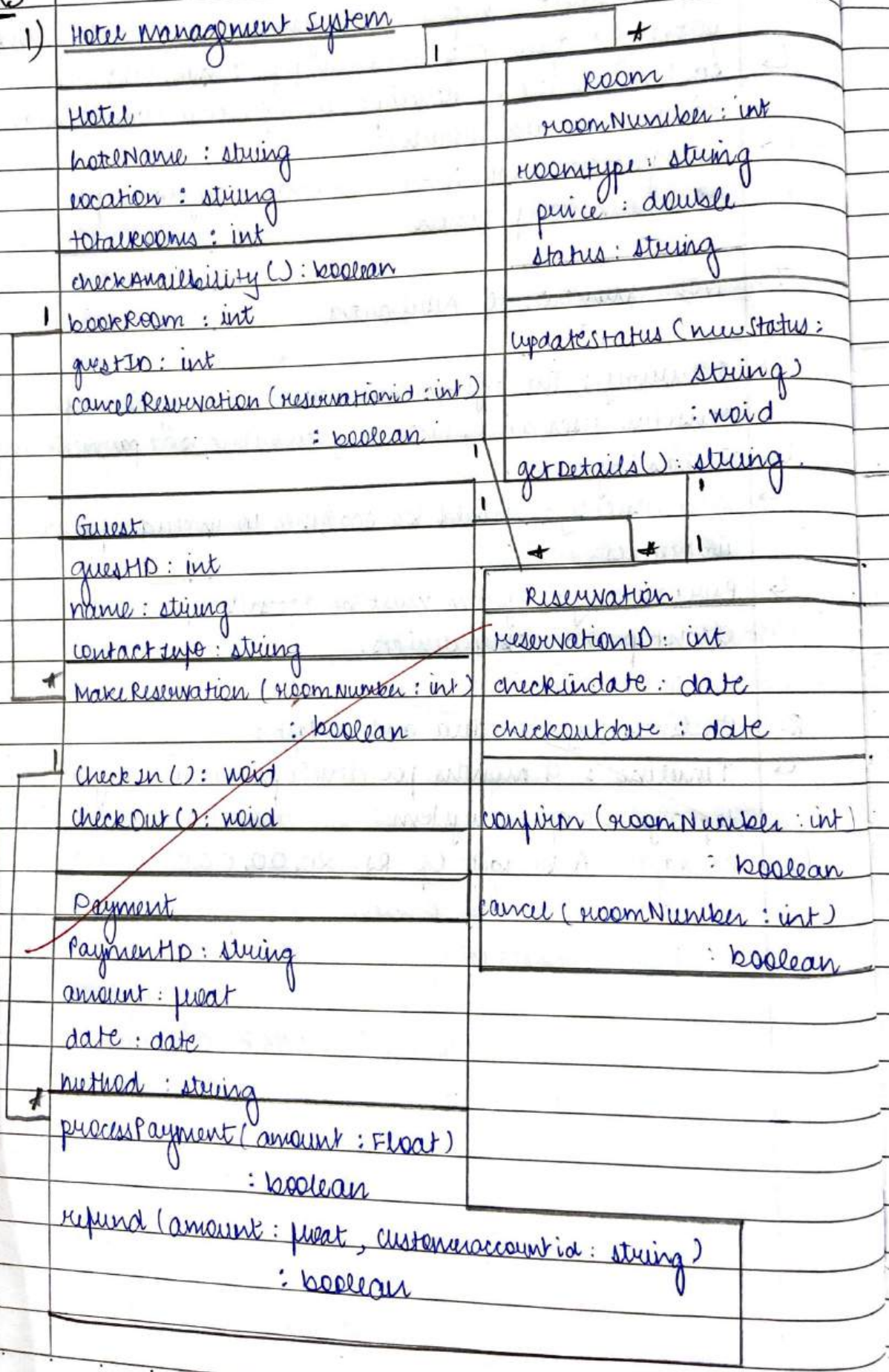
Preliminary Schedule and Budget :
Timeline : 9 months for development, testing and deployment of the system.
Budget : A budget of Rs. 20,00,000

CLASS DIAGRAM

Date __/__/__
Page

WEEK3

1) Hotel management system



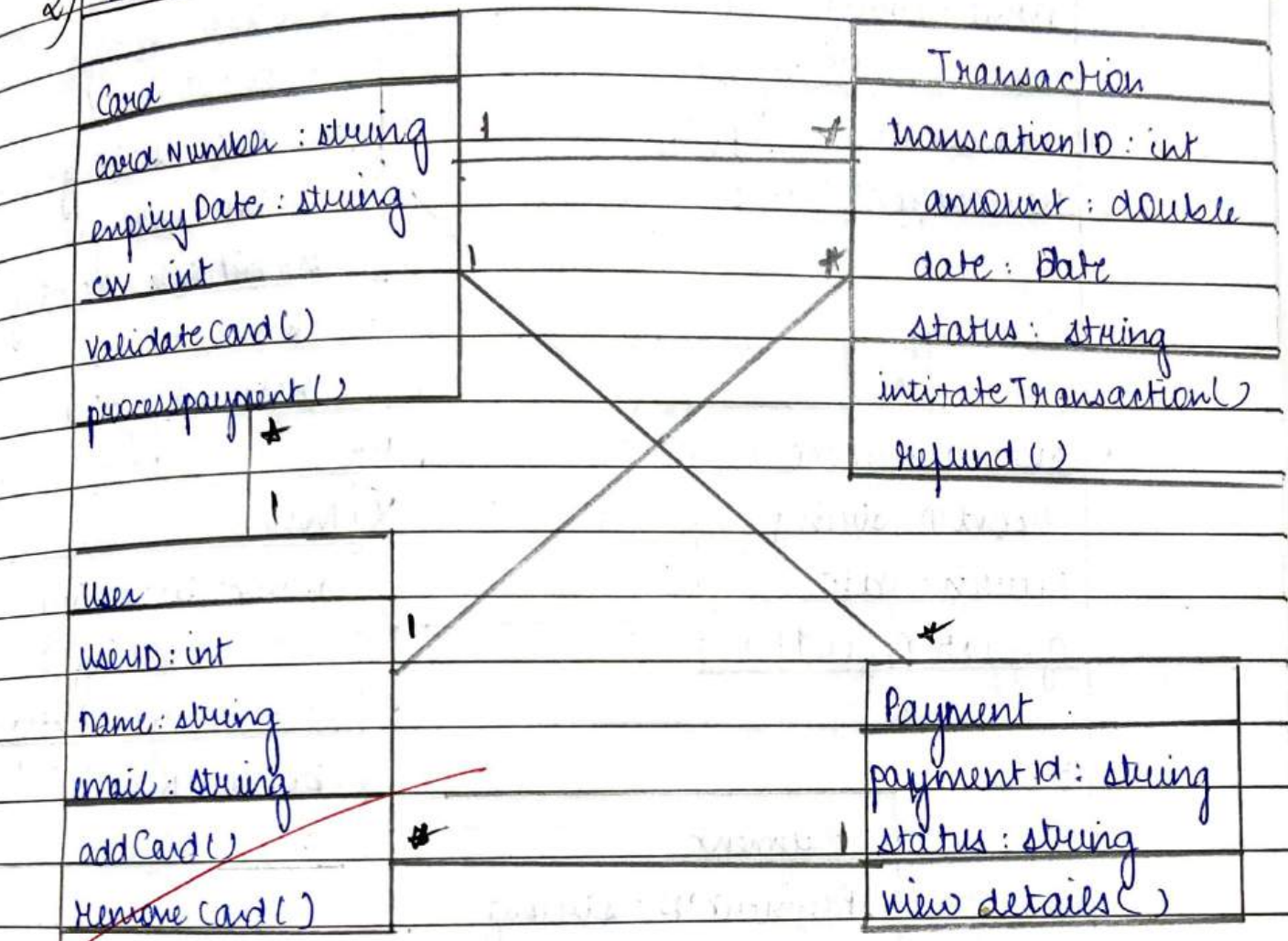
Class Diagram

2) Credit

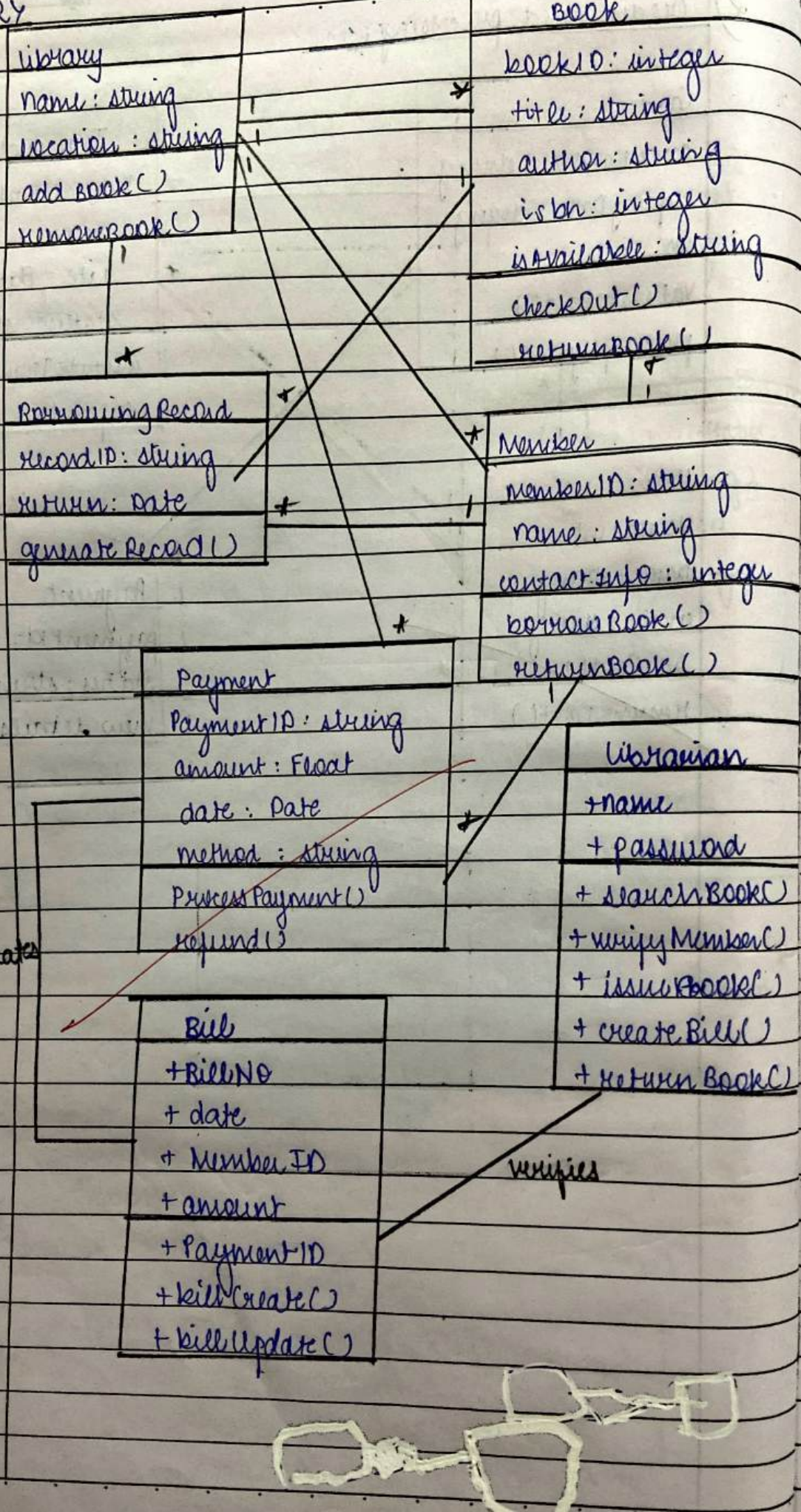
Date __/__/__
Page _____

Class Diagram

2) Credit card processing



3) LIBRARY

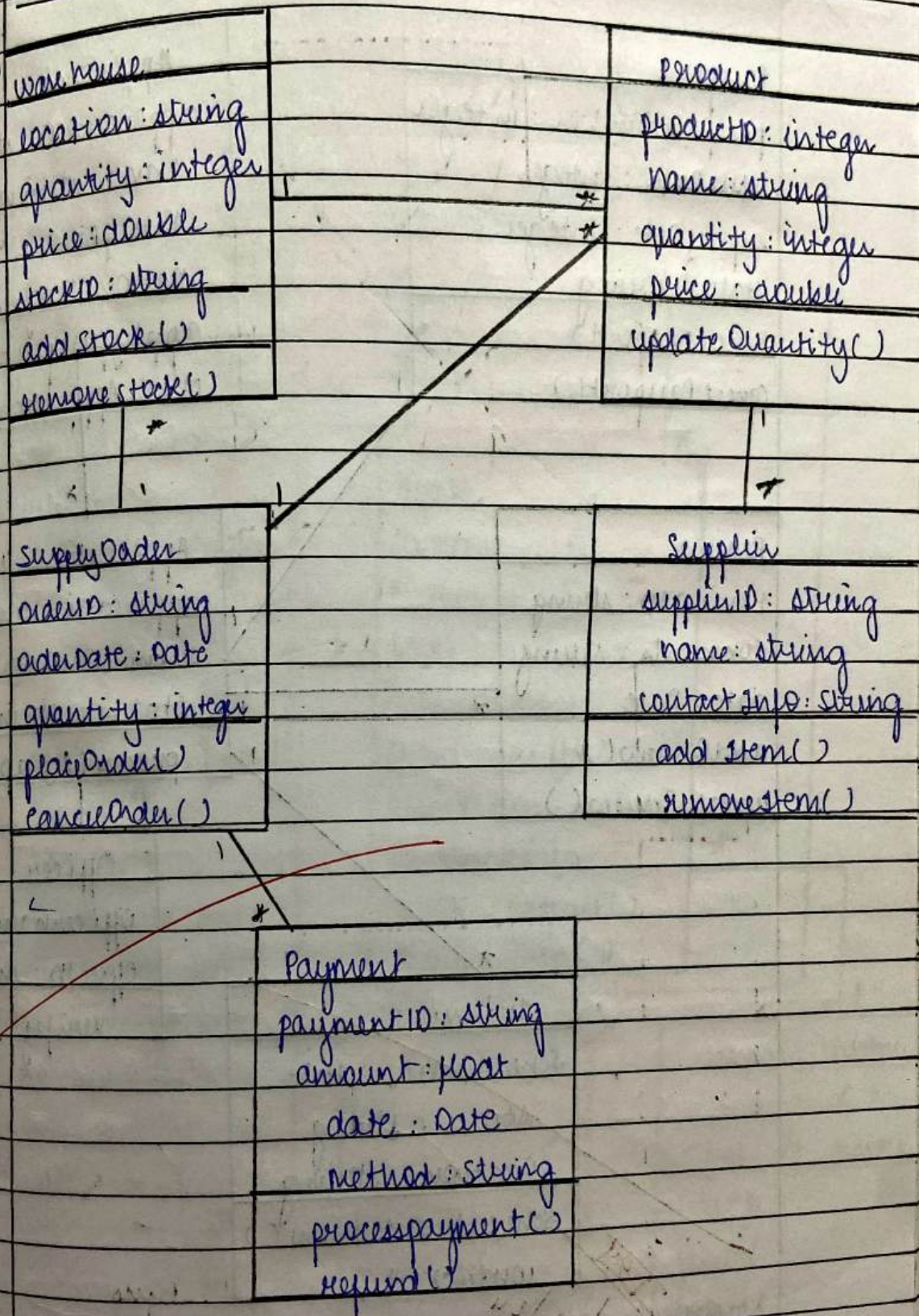

Library
name : string
location : string
addBook()
removeBook()
Book
bookID : integer
title : string
author : string
isbn : integer
isAvailable : string
checkOut()
returnBook()
BorrowingRecord
recordID : string
return : Date
generateRecord()
Member
memberID : string
name : string
contactInfo : integer
borrowBook()
returnBook()
Payment
PaymentID : string
amount : Float
date : Date
method : string
ProcessPayment()
refund()
Librarian
+name
+password
+searchBook()
+verifyMember()
+issueBook()
+createBill()
+returnBook()
creates
Bill
+BillNO
+date
+MemberID
+amount
+PaymentID
+billCreate()
+billUpdate()
verifies

class diagram

Date __/__/__
Page _____

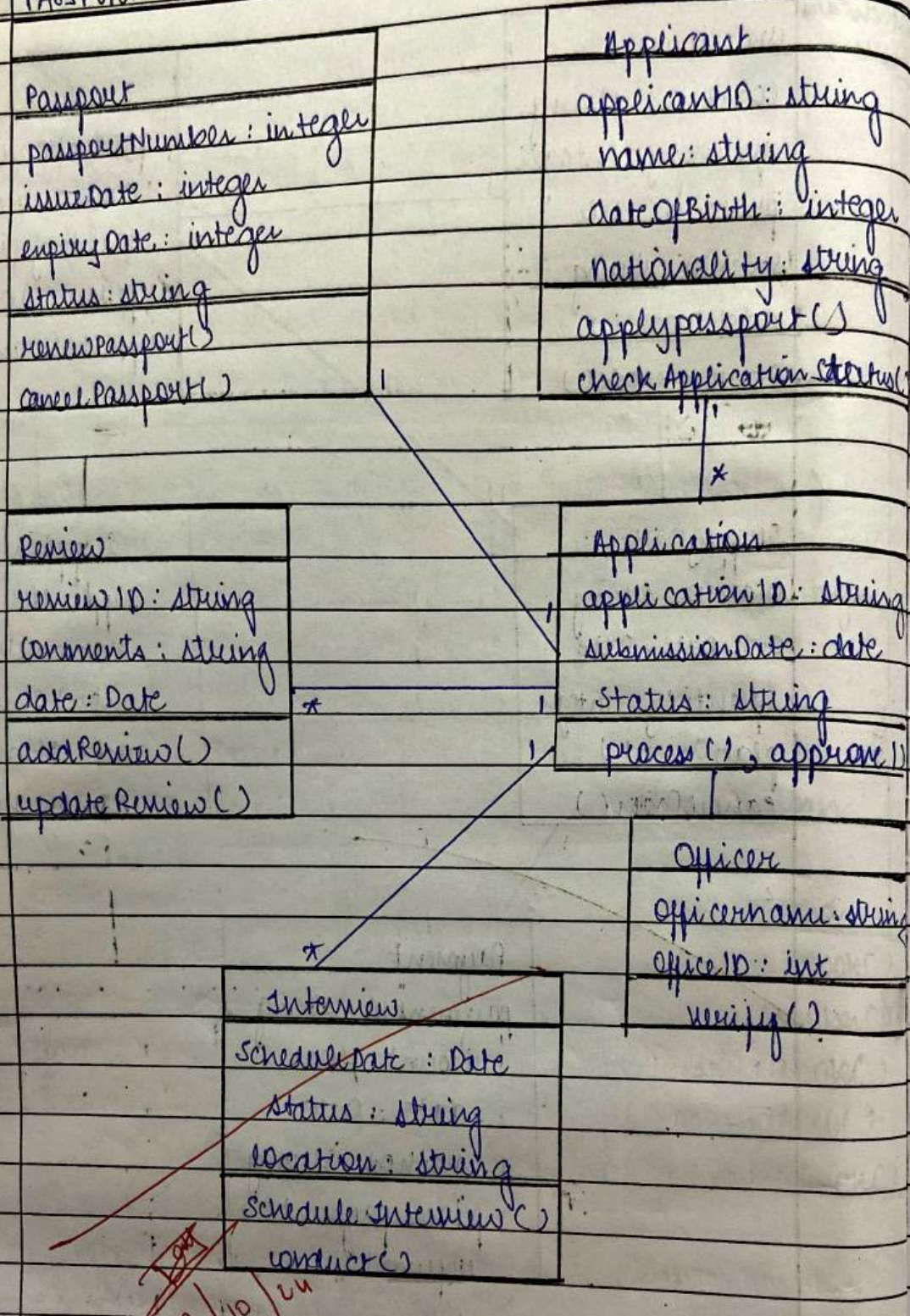
4) STOCK MAINTANCE SYSTEM

authenticate issue



Date __/__/__
Page _____

5) PASSPORT AUTOMATION SYSTEM

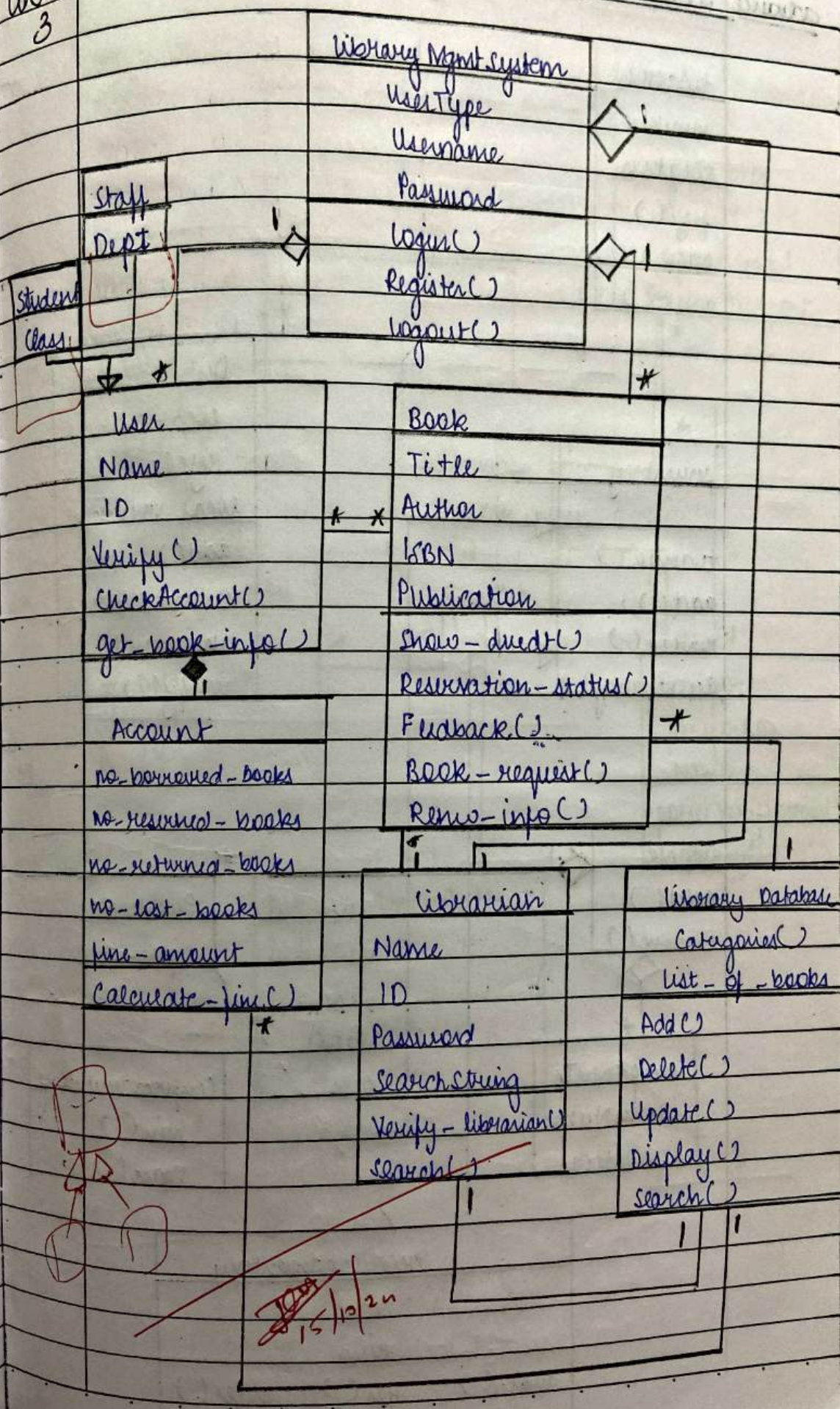


WEEK 3

ADVANCED CLASS DIAGRAMS

LIBRARY MGMT SYSTEM

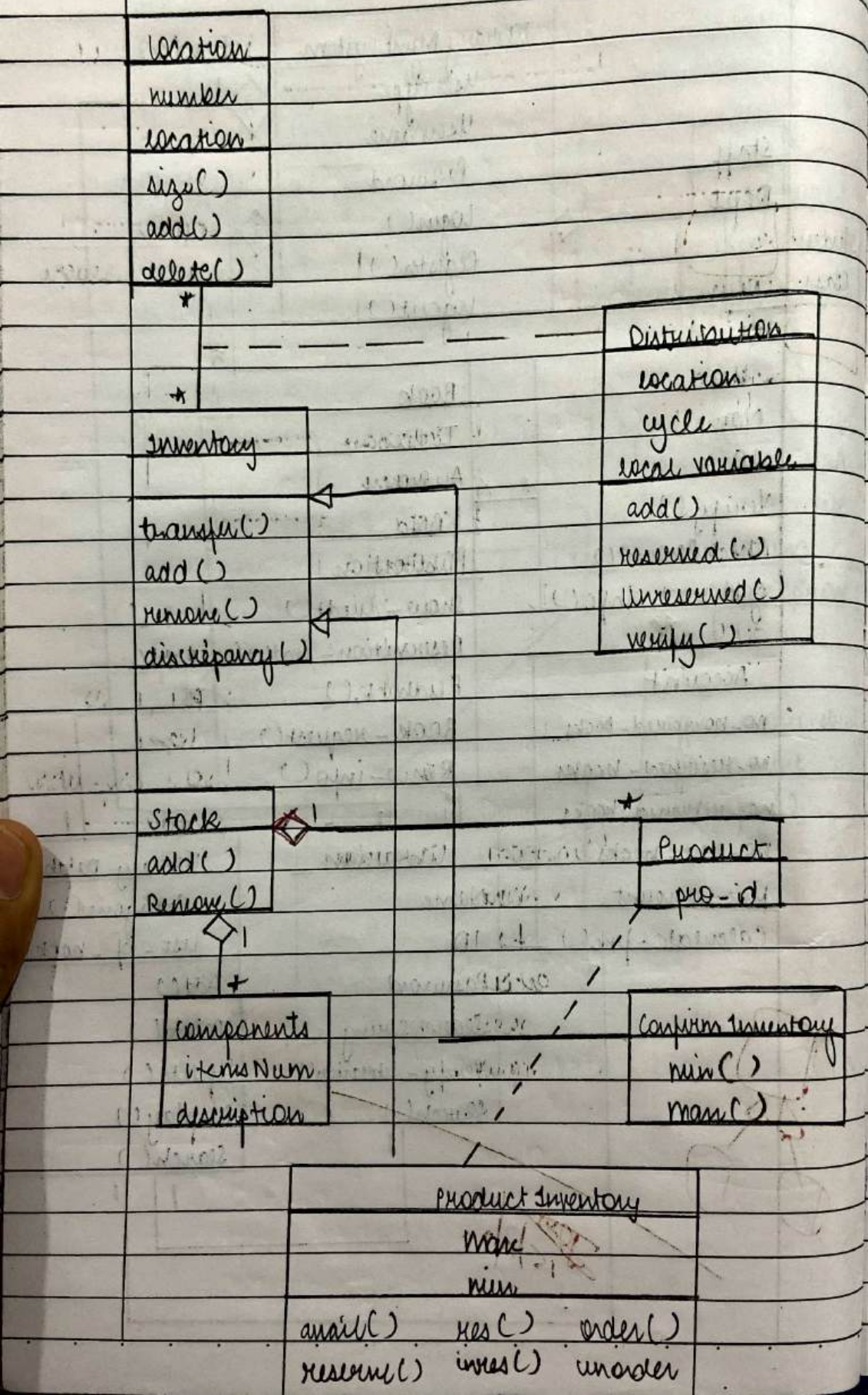
Date __/__/__
Page



advanced class diagrams

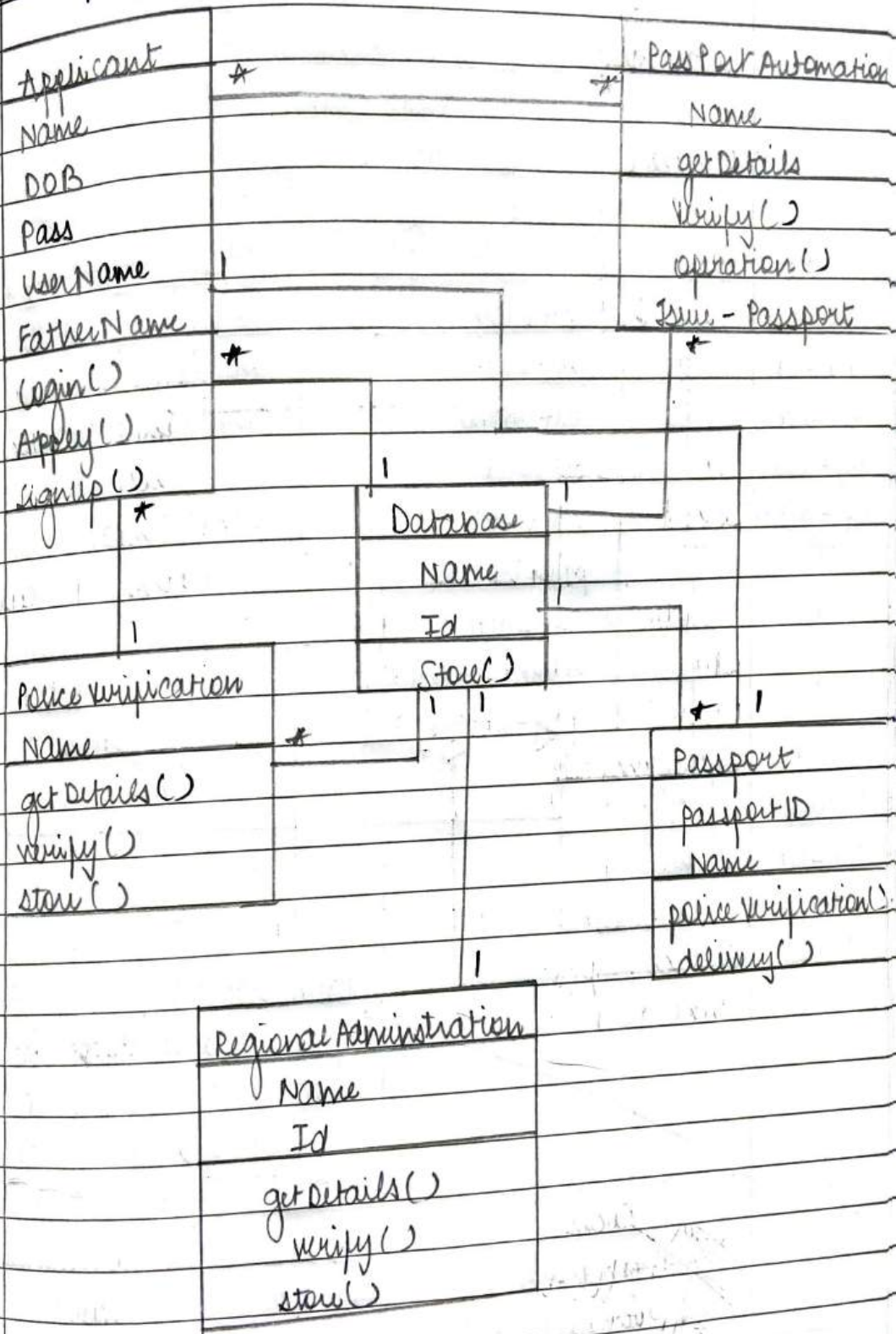
Date __/__/__
Page

advance STOCK MAINTAINANCE SYSTEM



Date __/__/__
Page _____

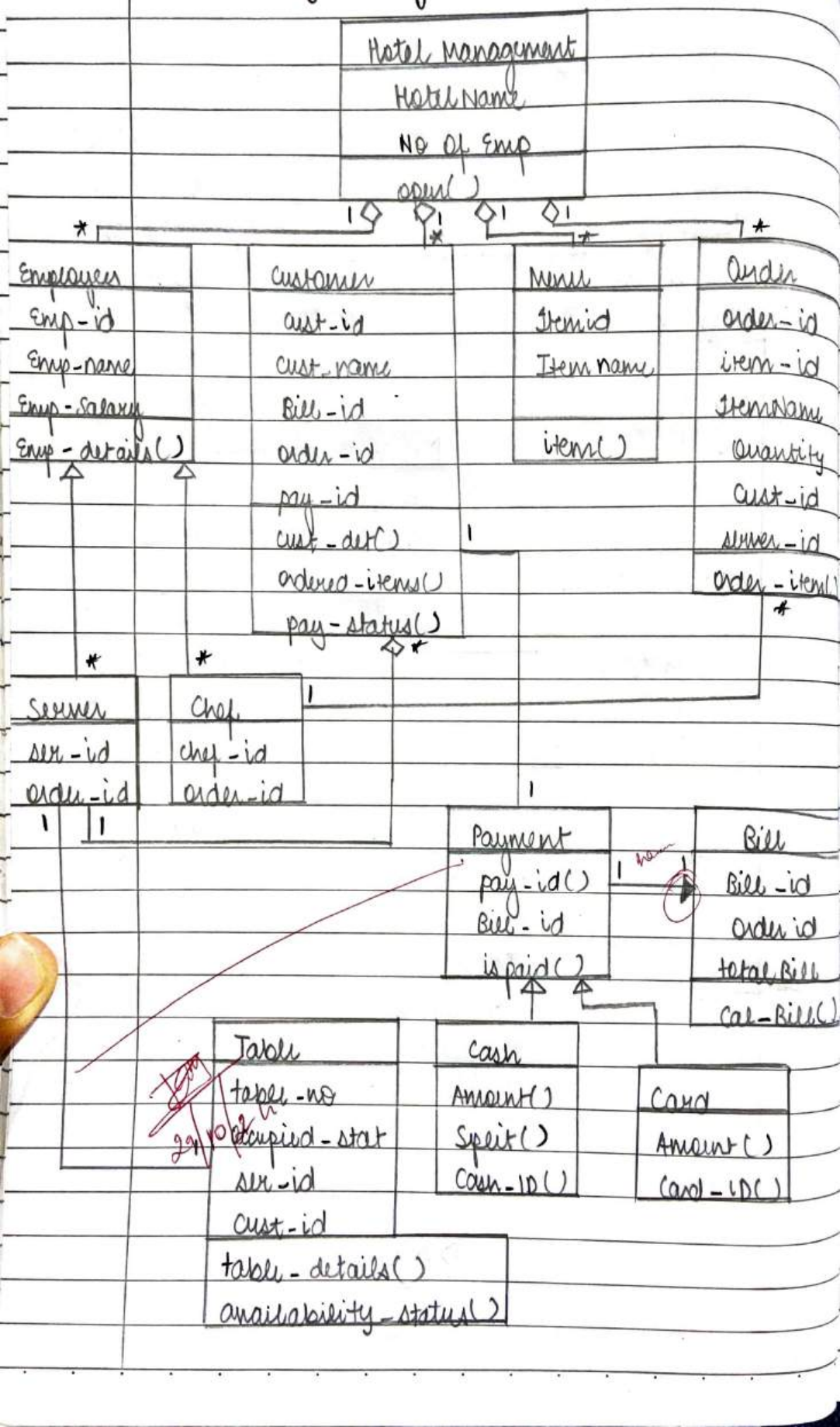
Passport automation system :



class diagrams

Date __/__/__
Page

advanced Hotel Management System

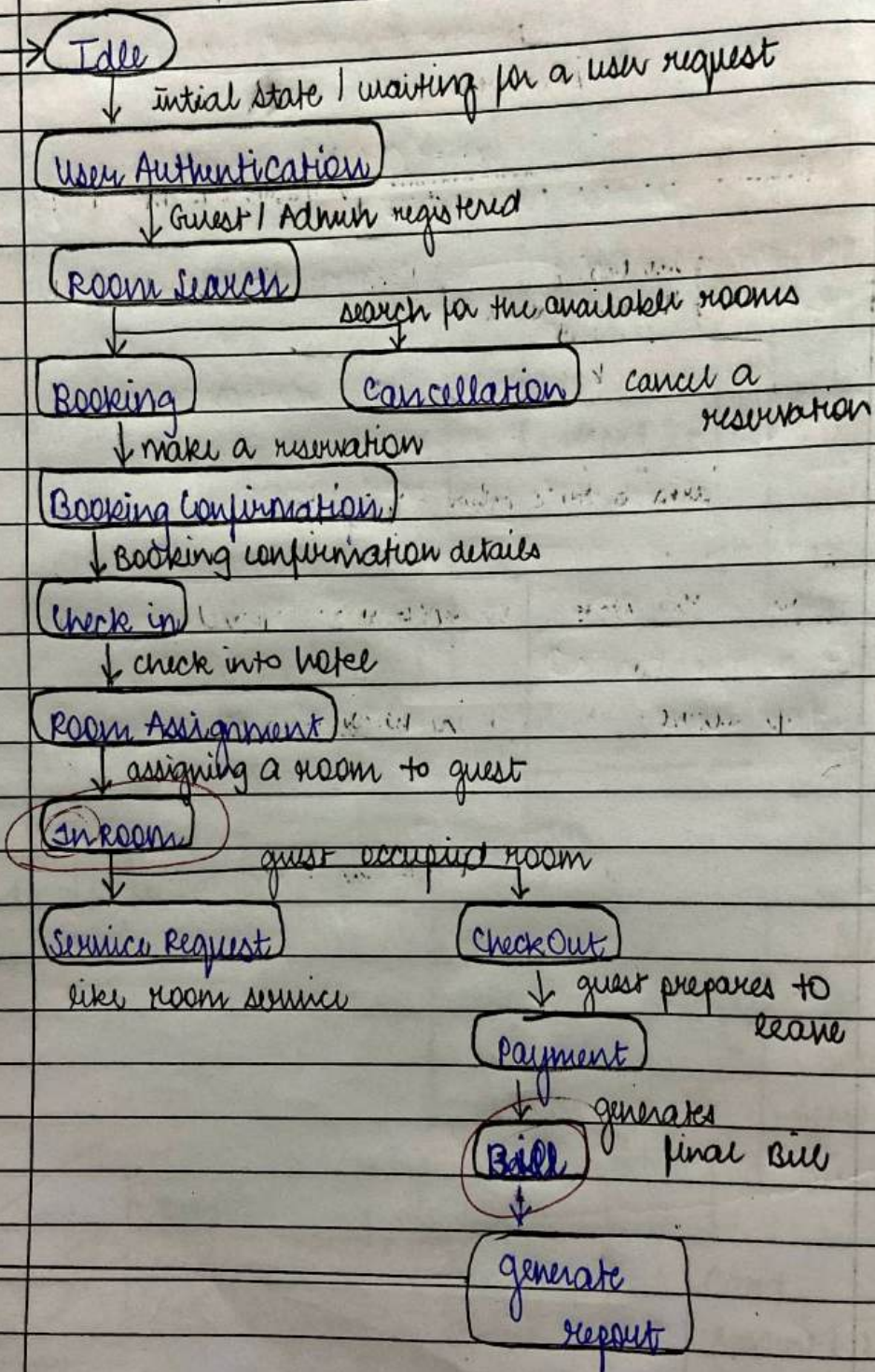


Date __/__/__
Page

STATE DIAGRAMS

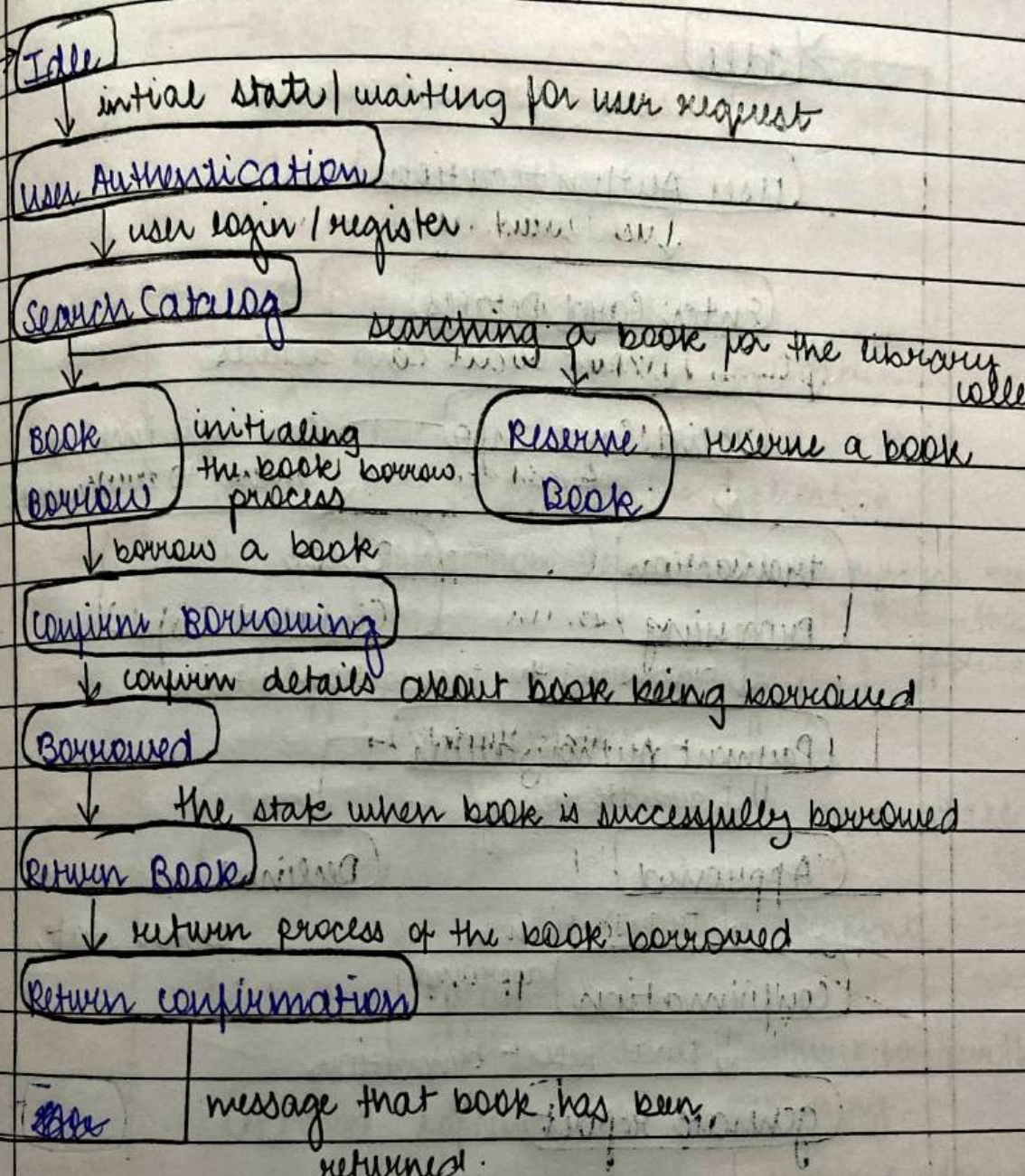
WEEK 5

1) Hotel management system

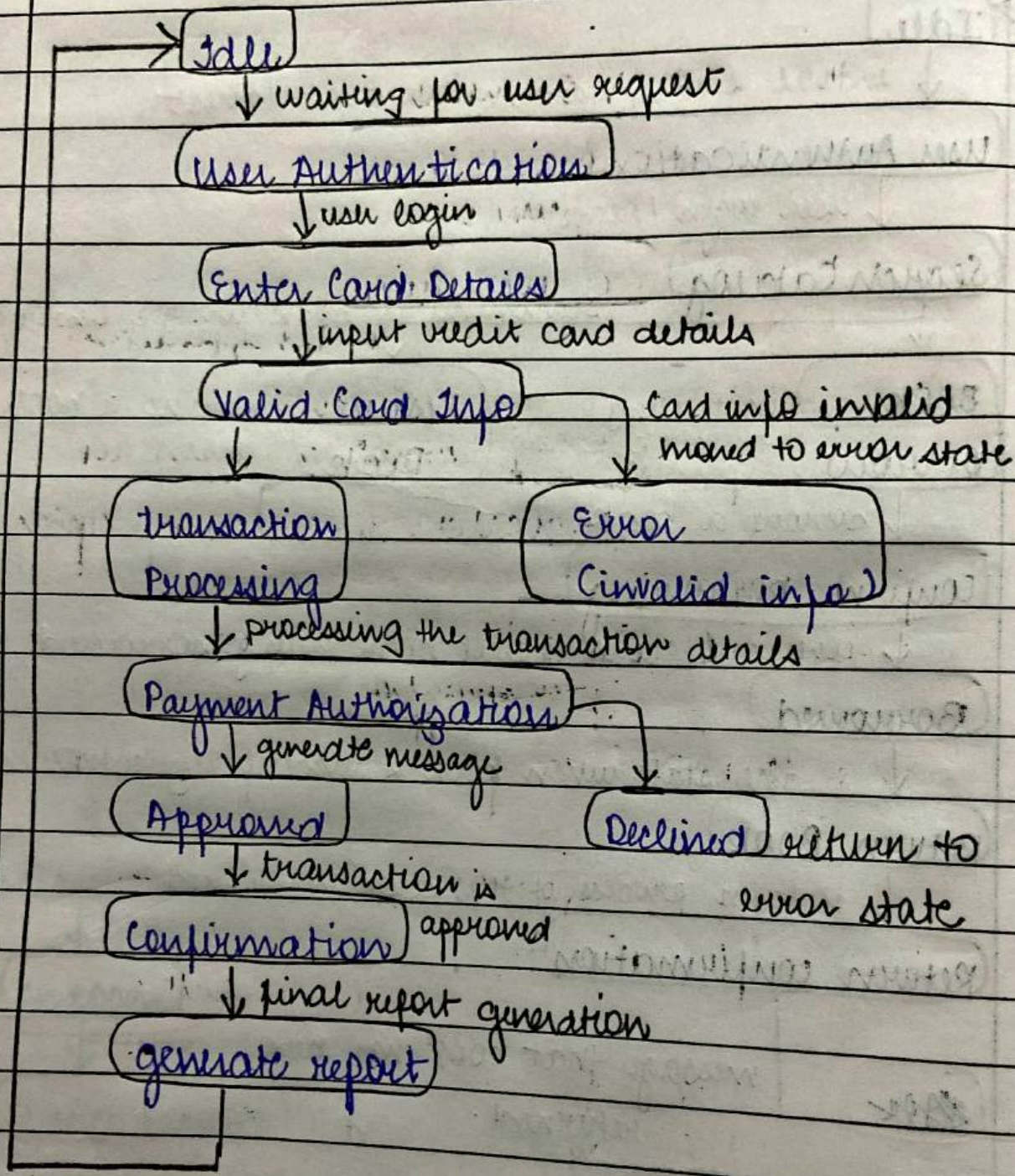


Date __/__/__
Page _____

2) → State Diagram for library Mgmt System

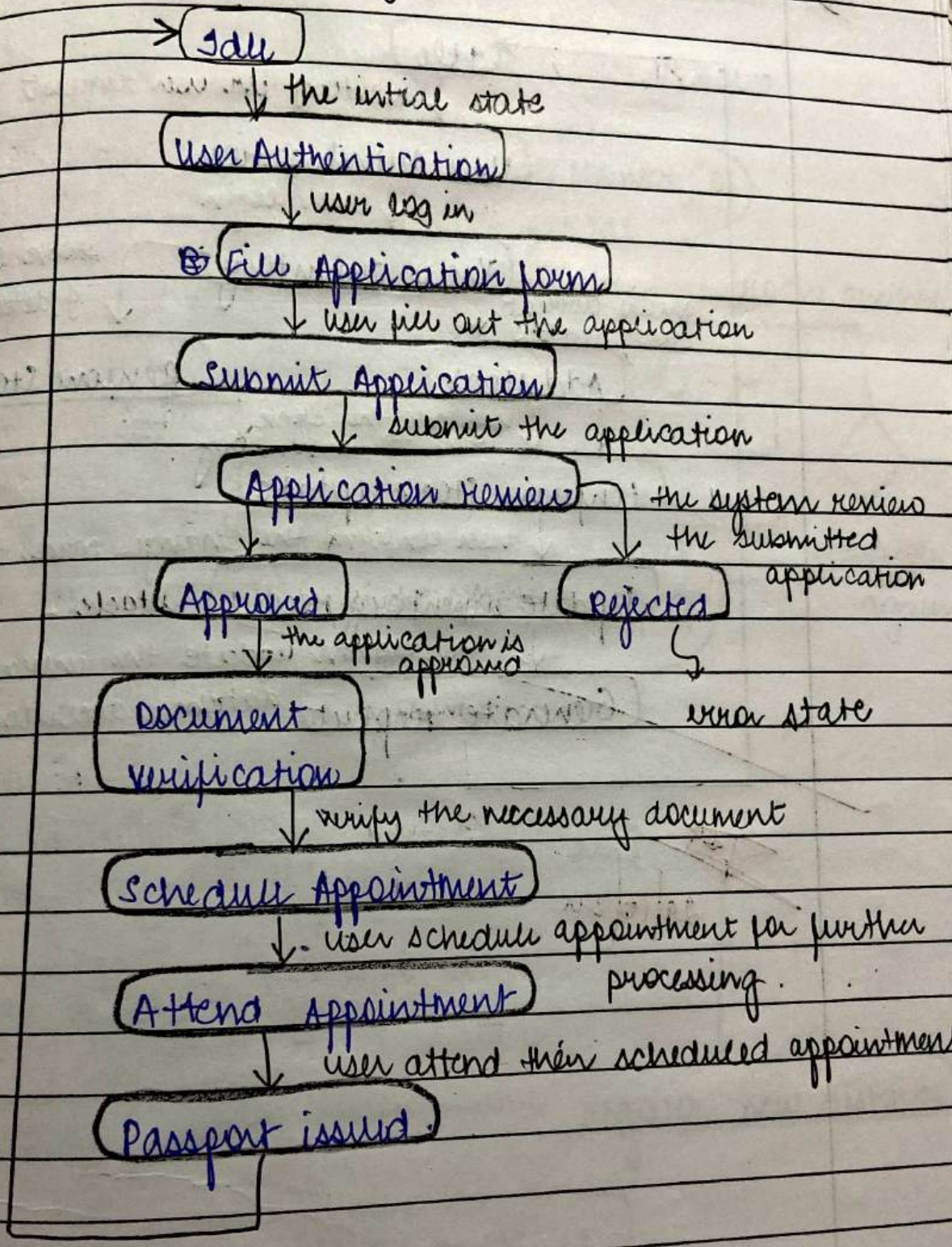


3) Credit card Processing



Date __/__/__
Page _____

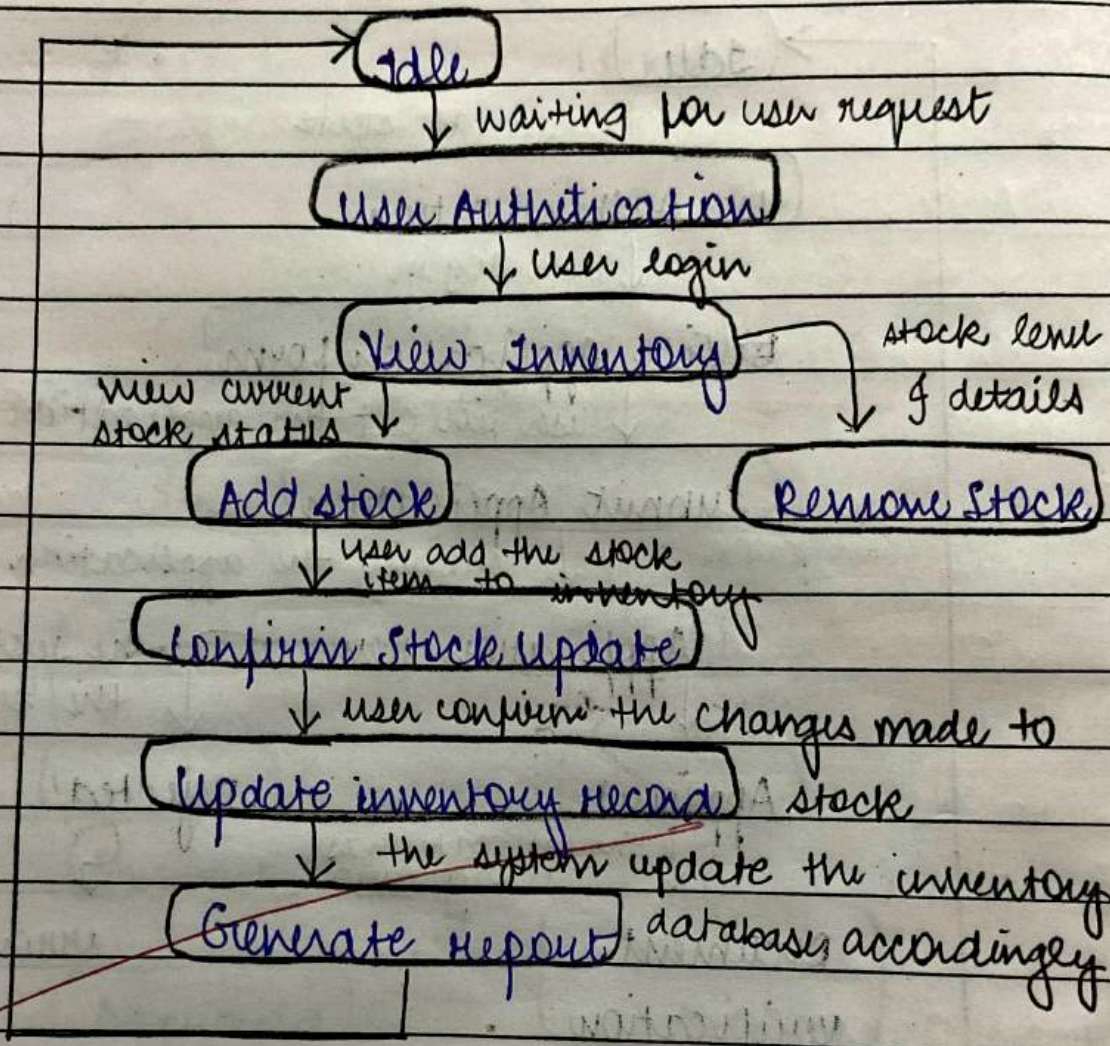
4) Passport Automation system



State diagram

Date __/__/__
Page

5) Stock Maintance system

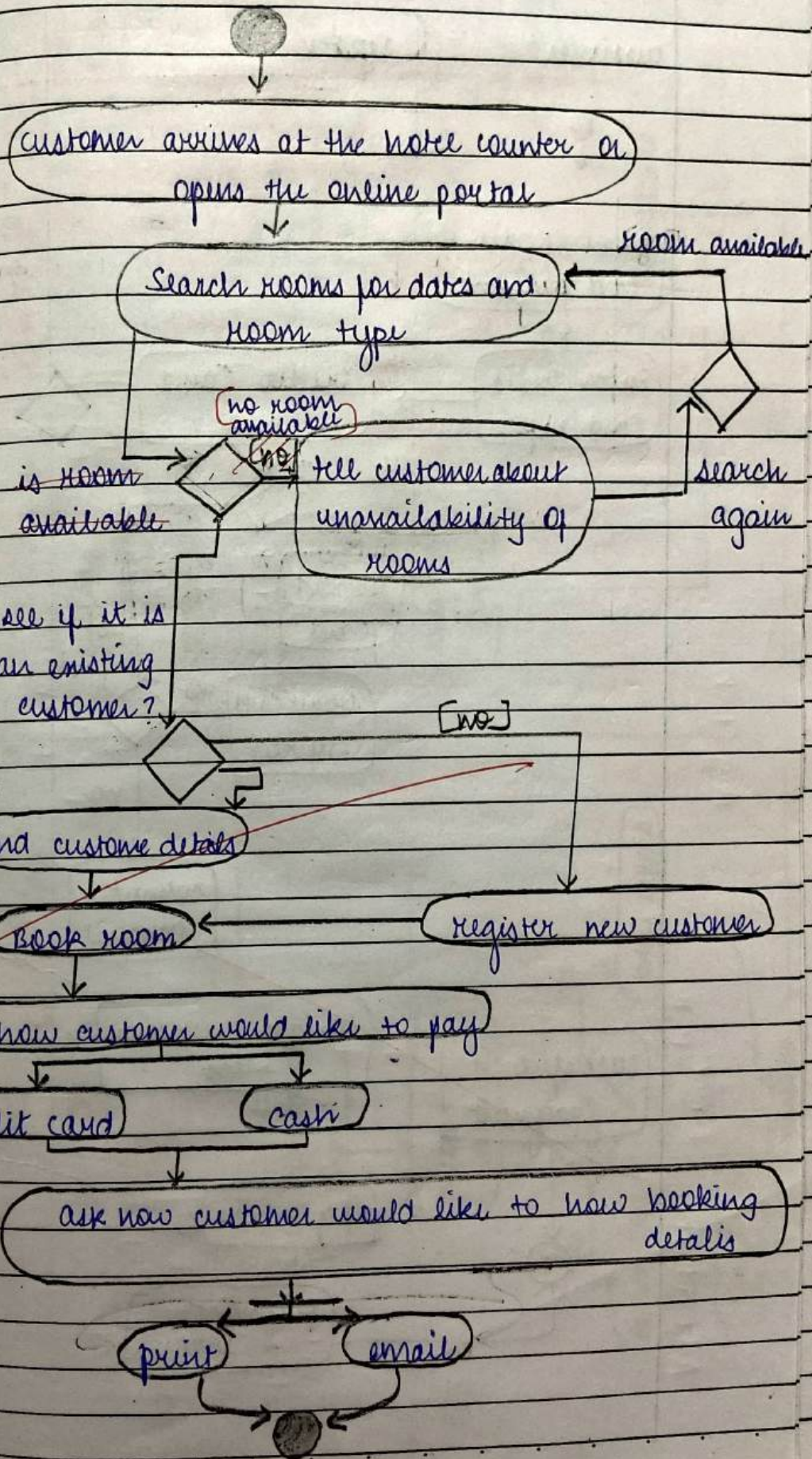


29/10/24

ACTIVITY DIAGRAMS

Date __/__/__
Page

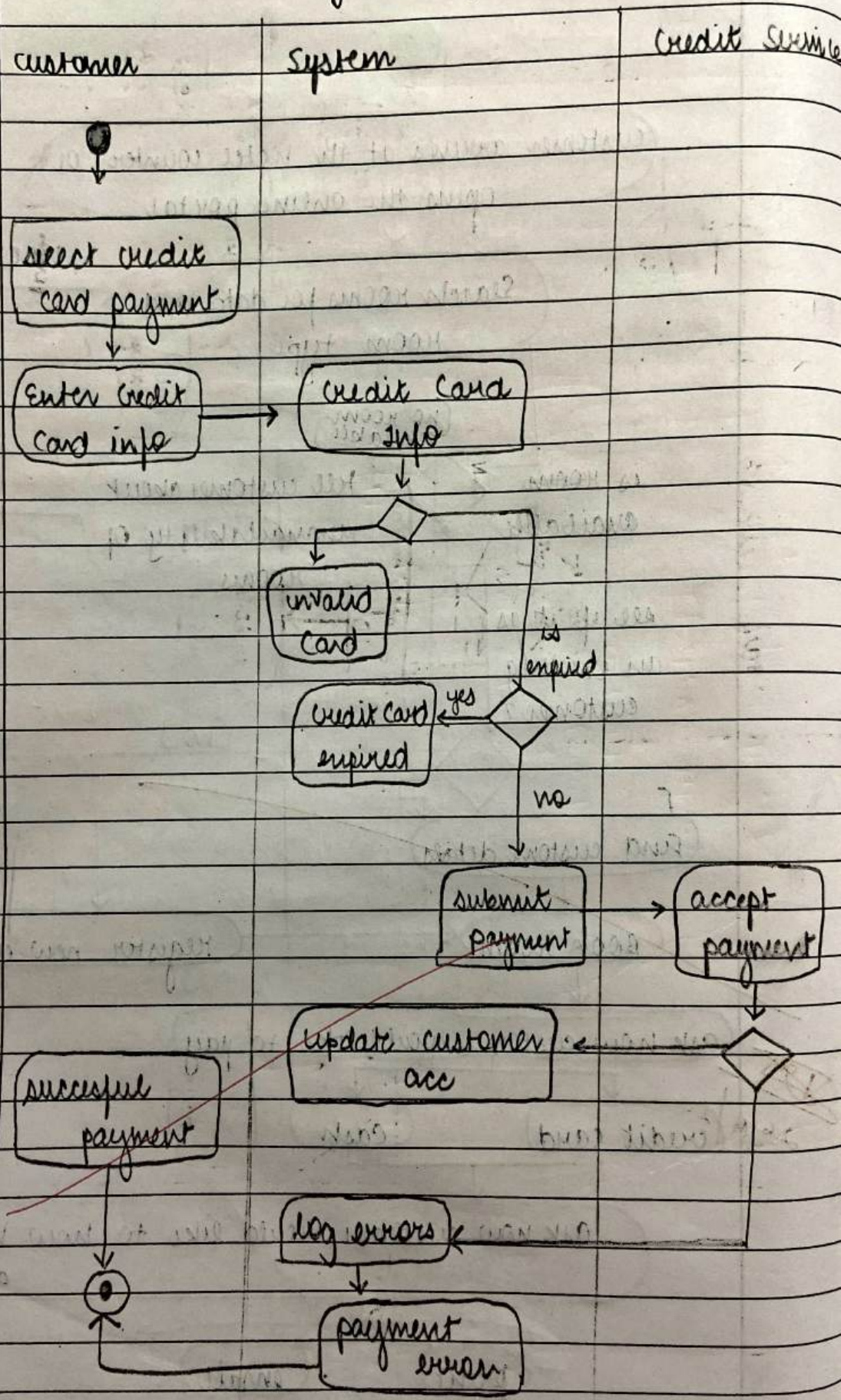
1) ACTIVITY DIAGRAM FOR HOTEL MANAGEMENT SYSTEM



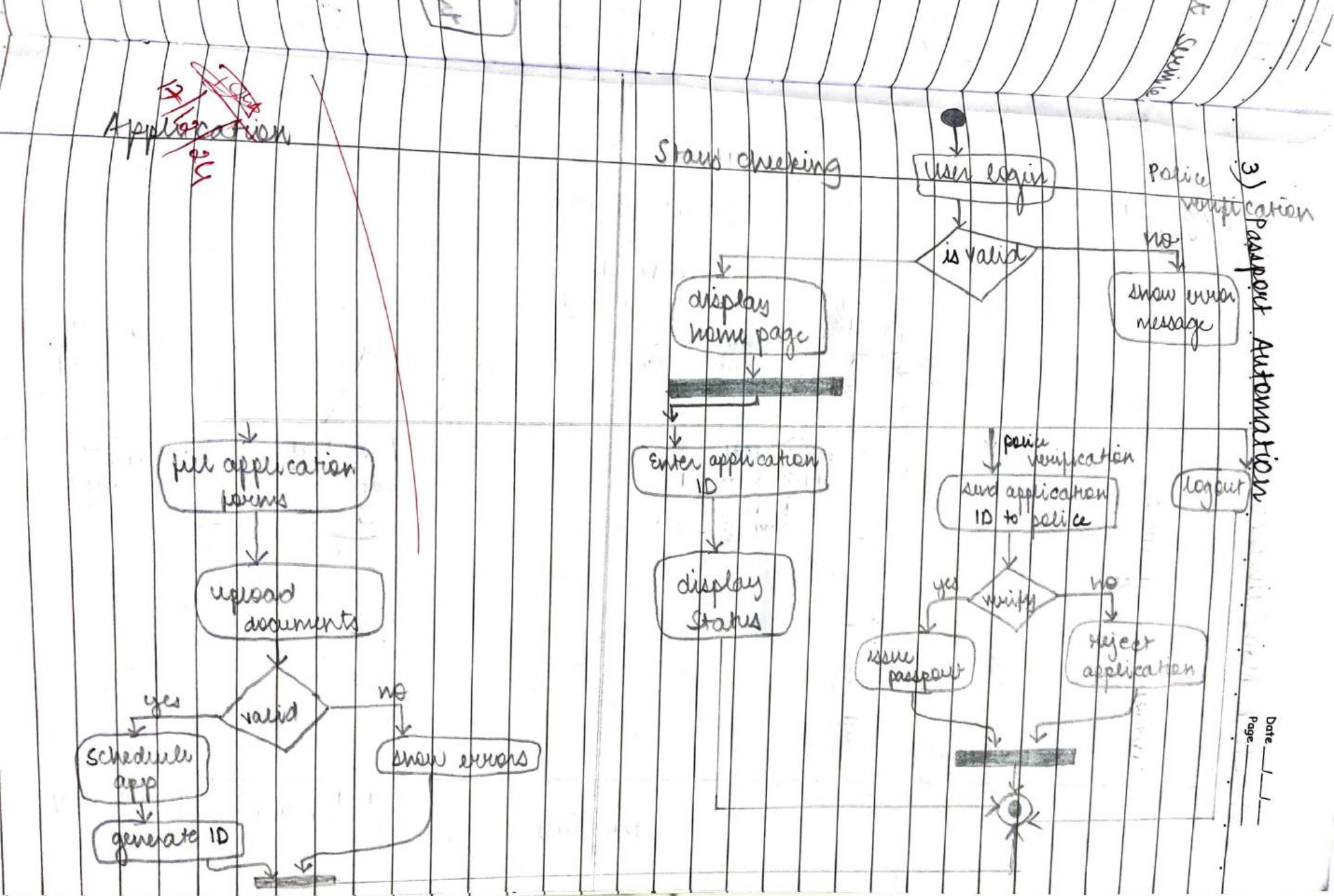
Date
Page

activity diagram:

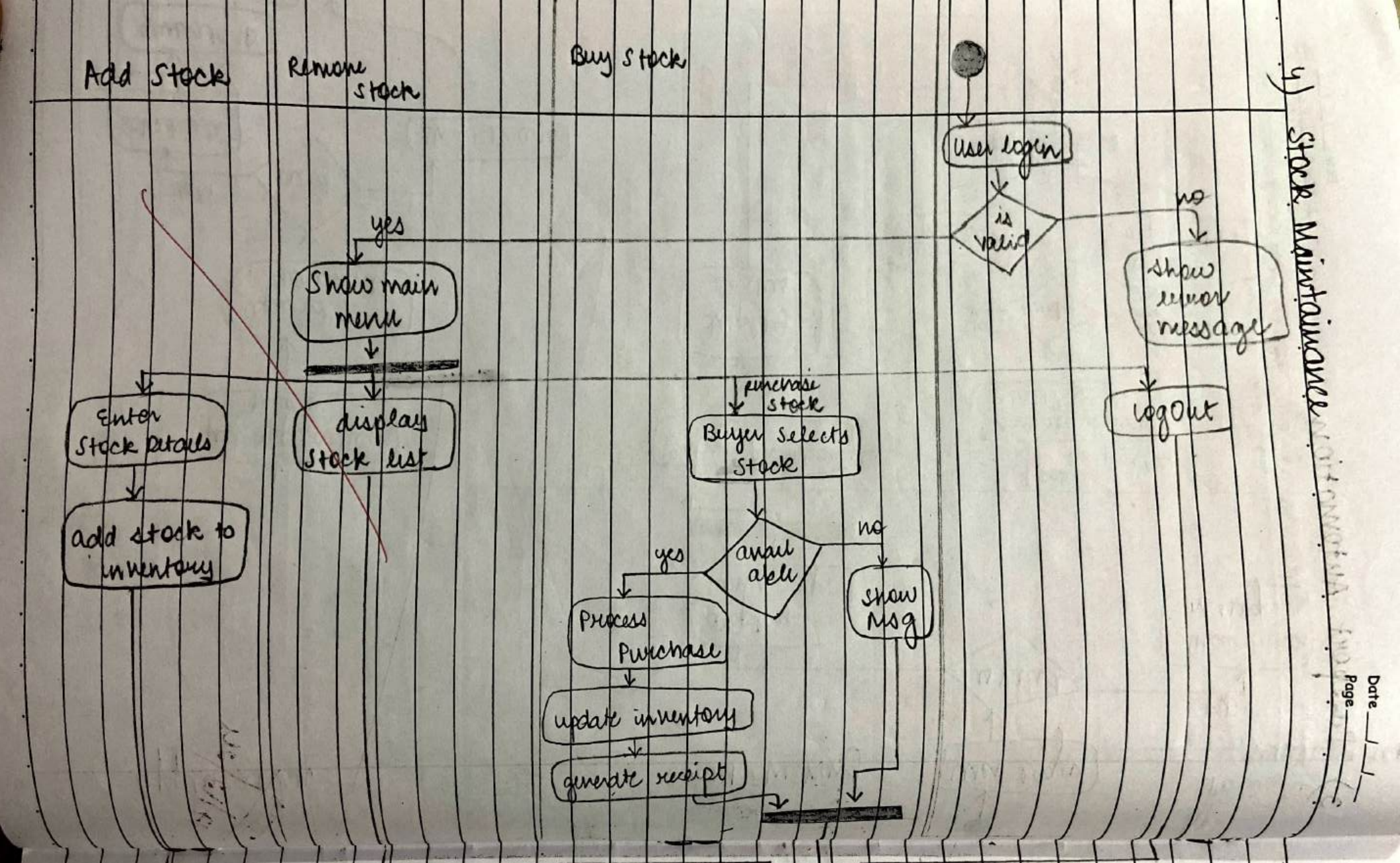
2) Credit Card Management system:



3) Passport Automation

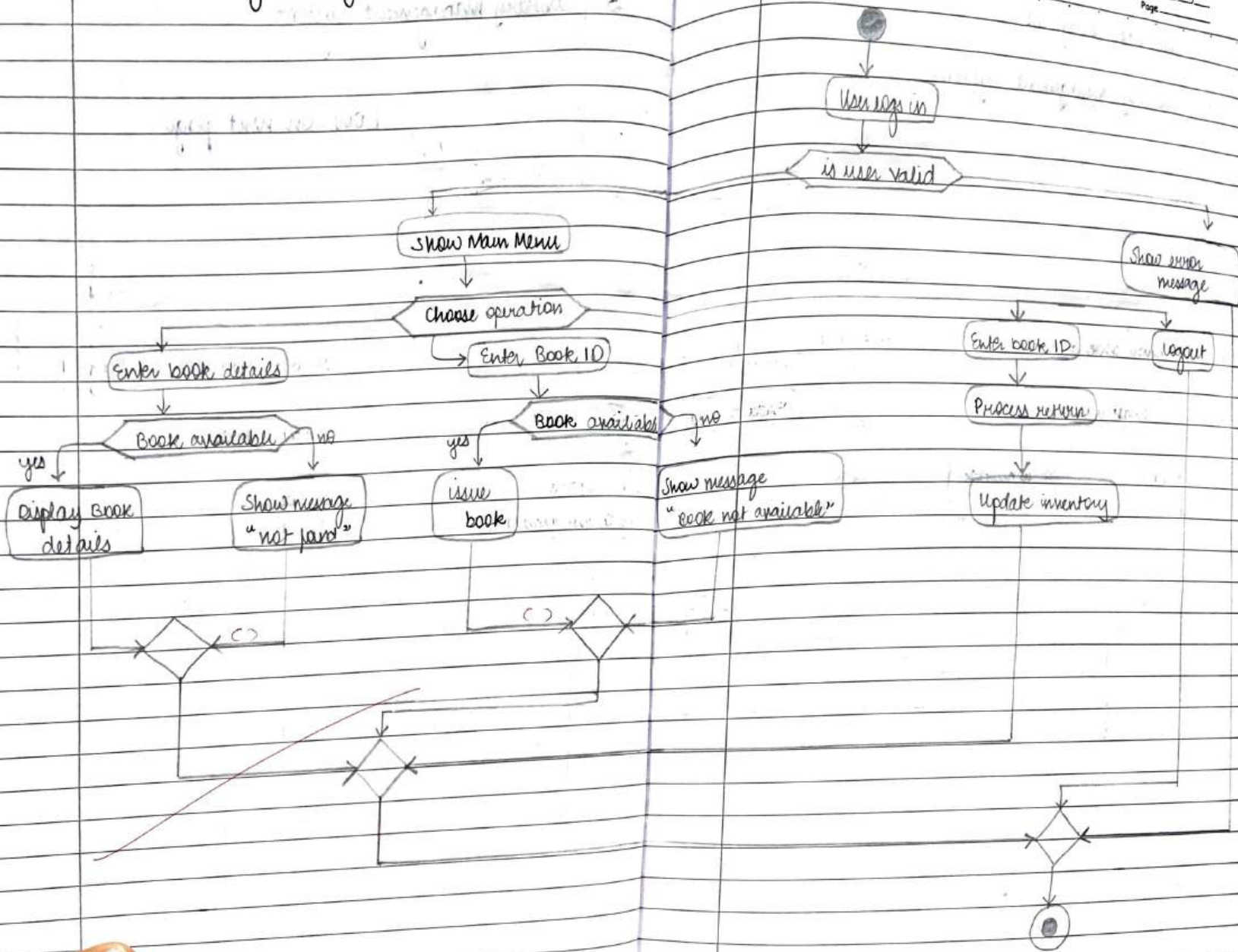


4) Stock Maintainance



Date __/__/__
Page

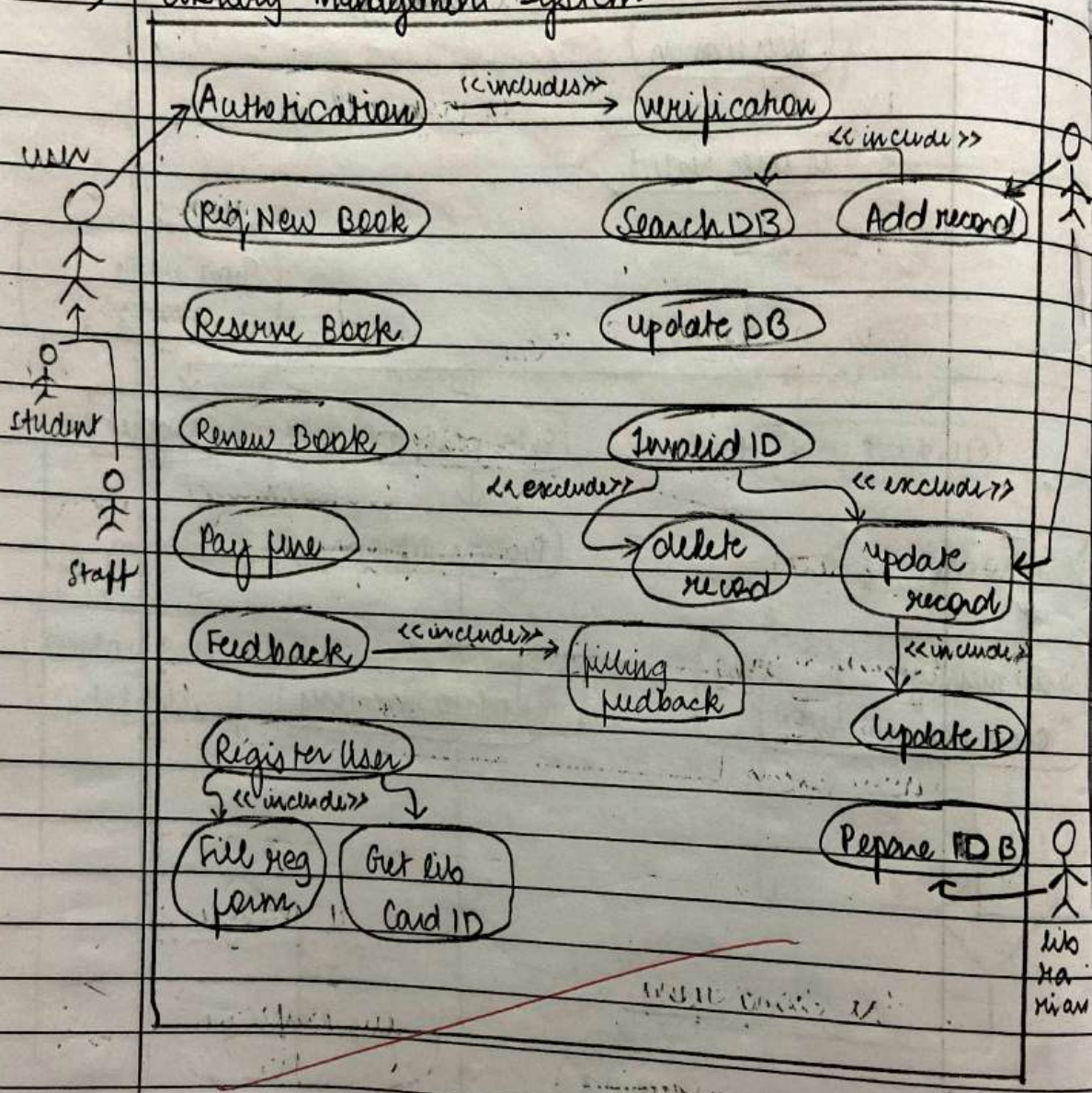
5) Library Management System



Date __/__/__
Page _____

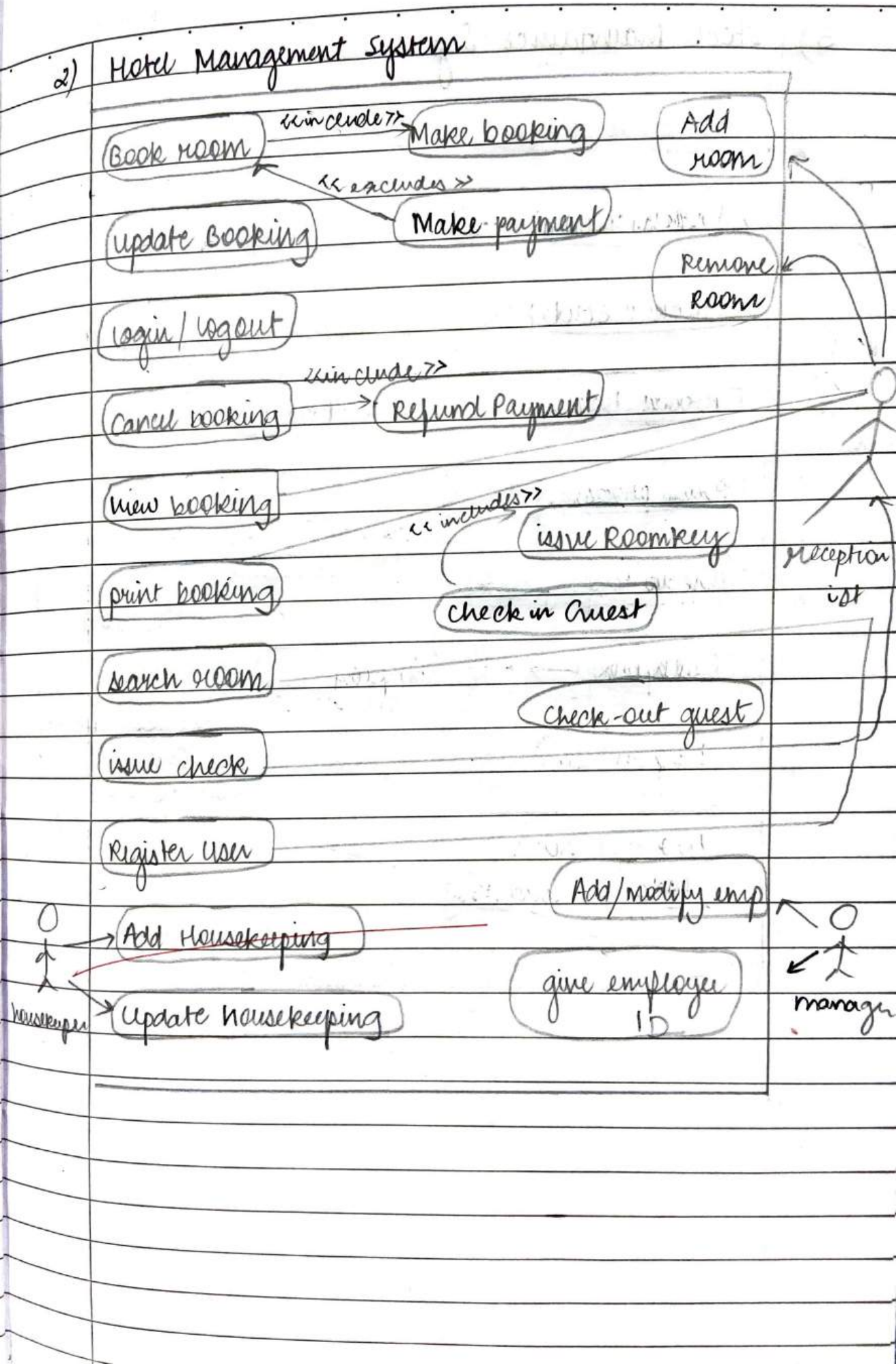
Use Case Diagrams

1) Library Management System



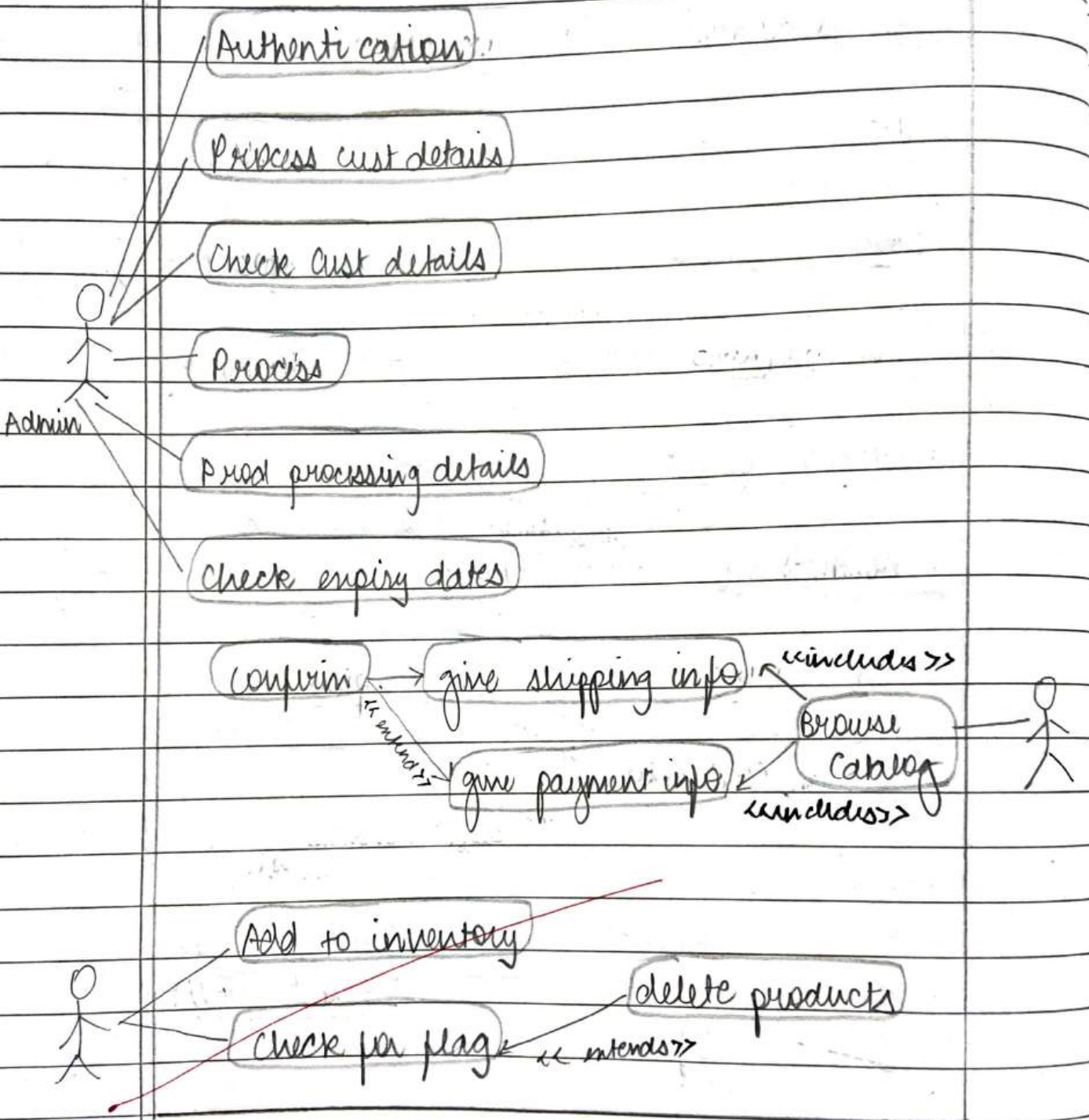
Date __/__/__
Page _____

2) Hotel Management System



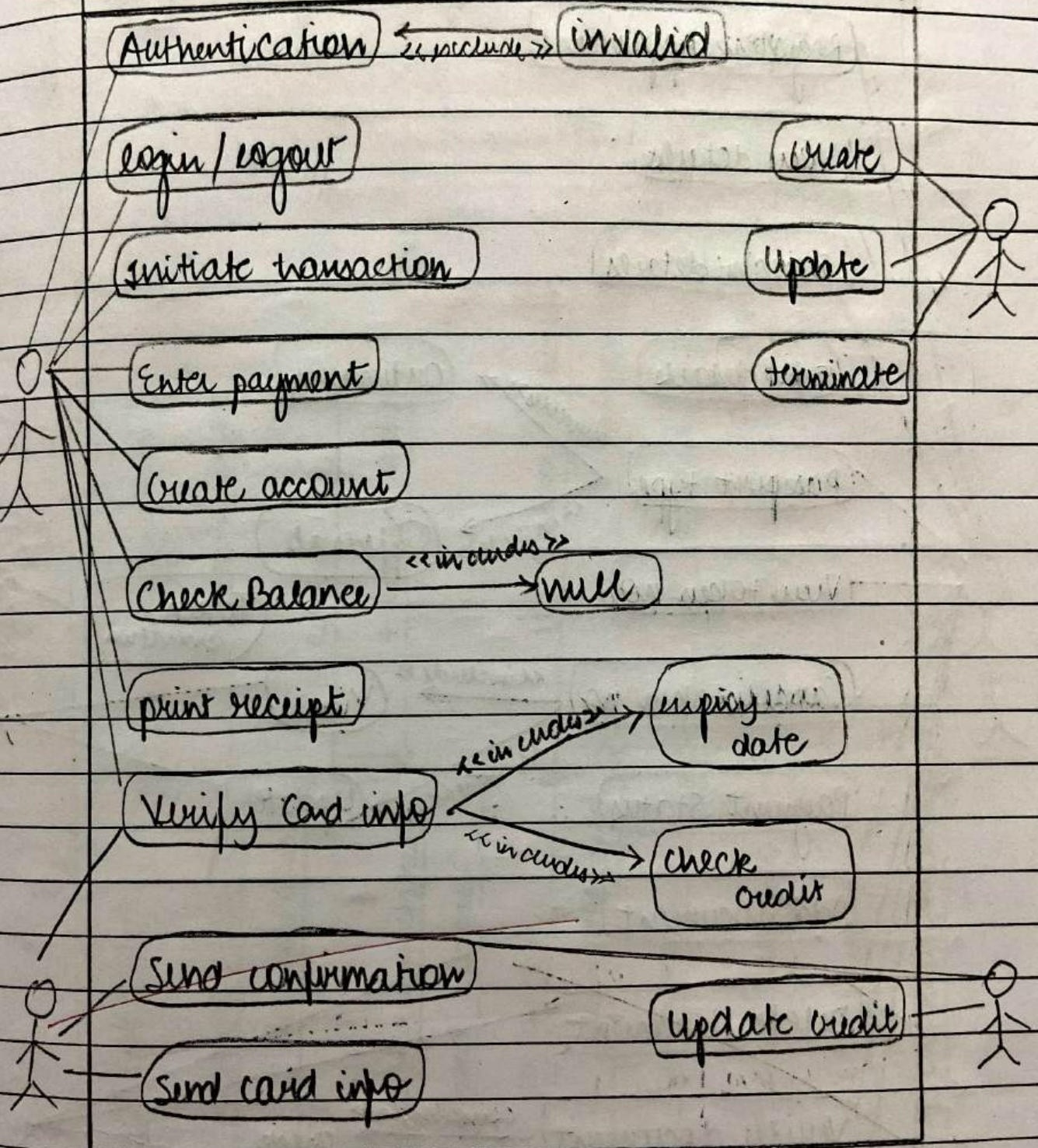
Date __/__/__
Page _____

3) Stock Maintaince System



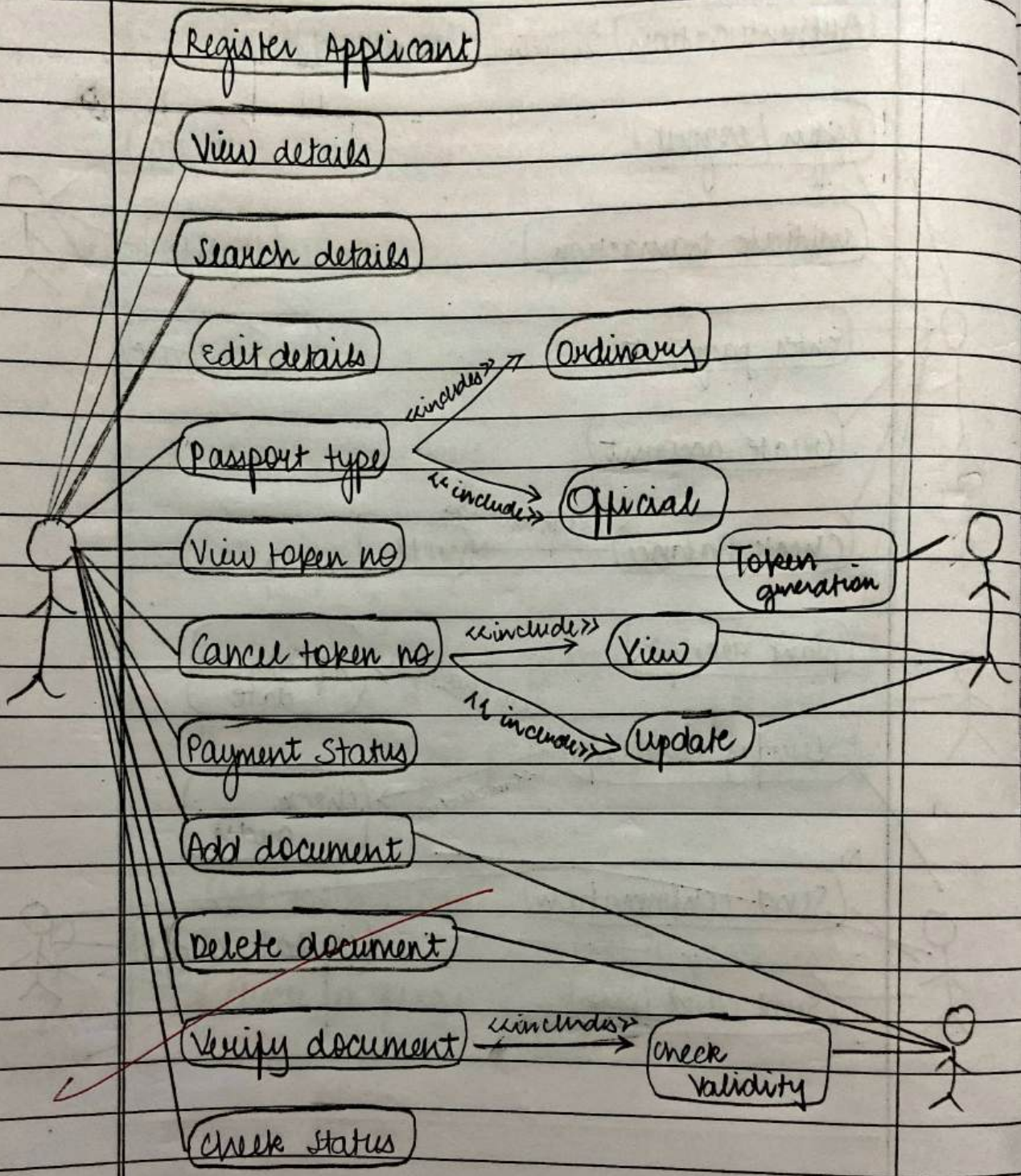
Date __/__/__
Page _____

4) Credit card processing System



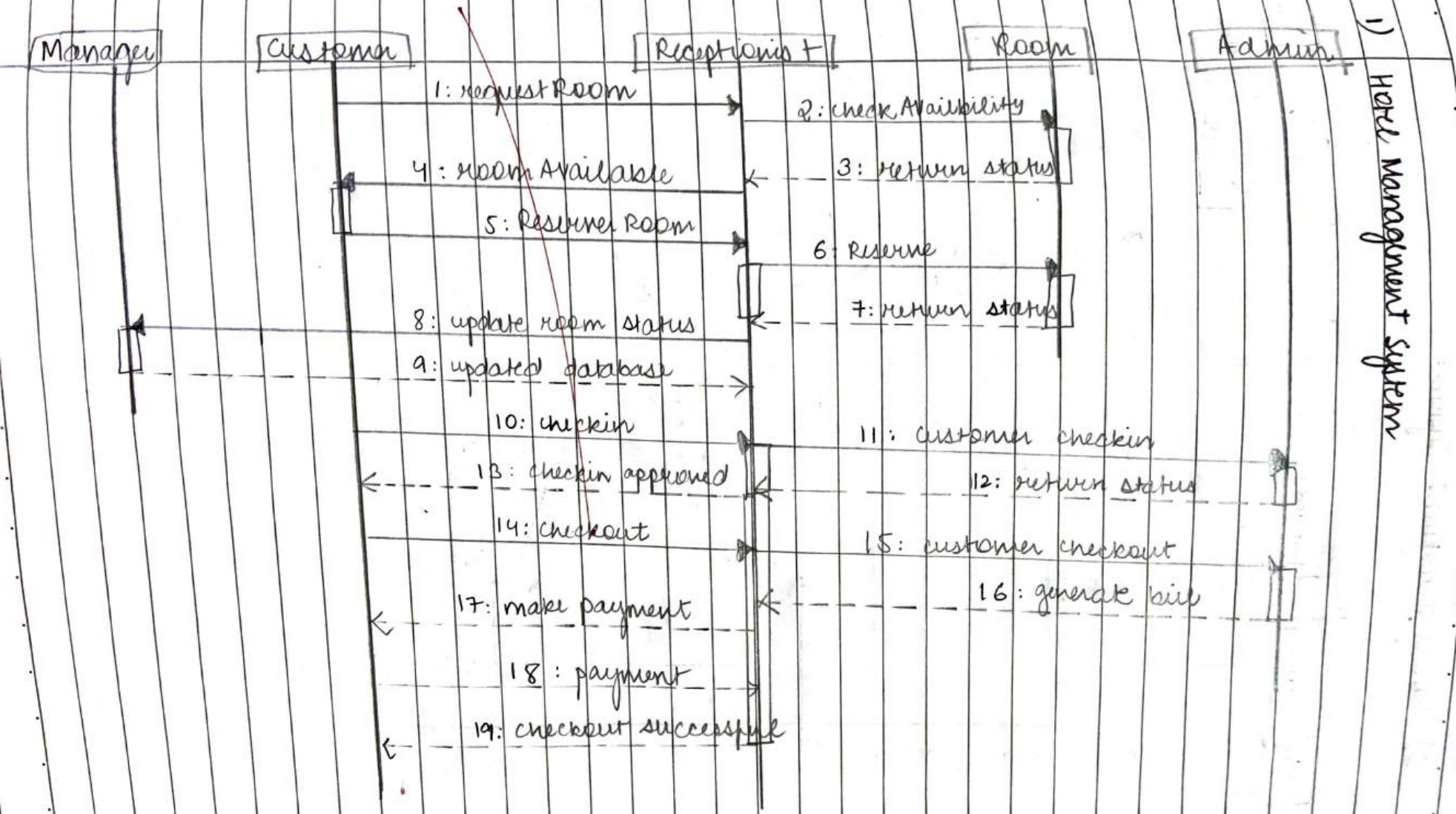
Date __/__/__
Page _____

5) Passport Automation System



SEQUENCE DIAGRAM

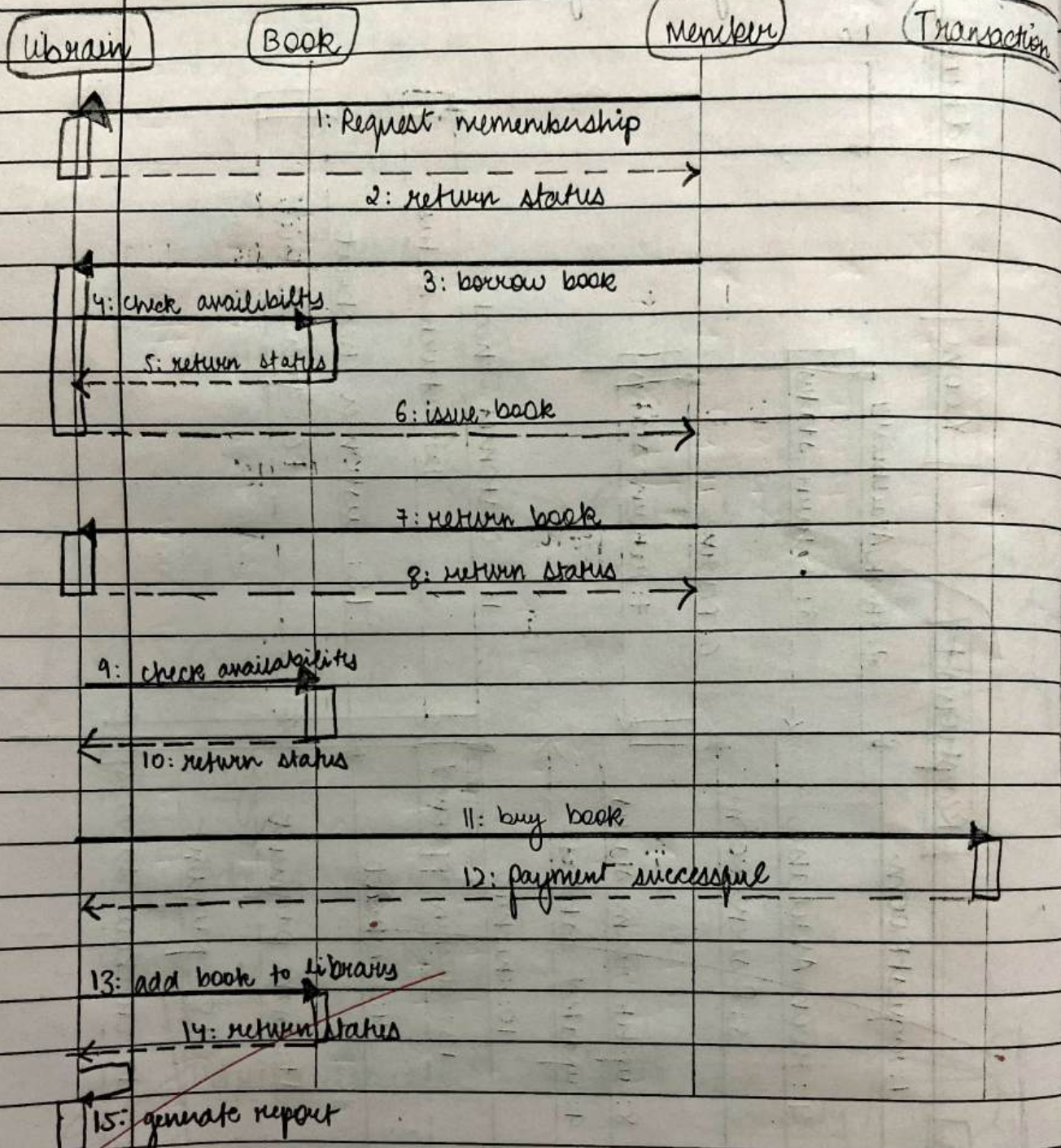
1) Hotel Management System



Date __/__/__
Page

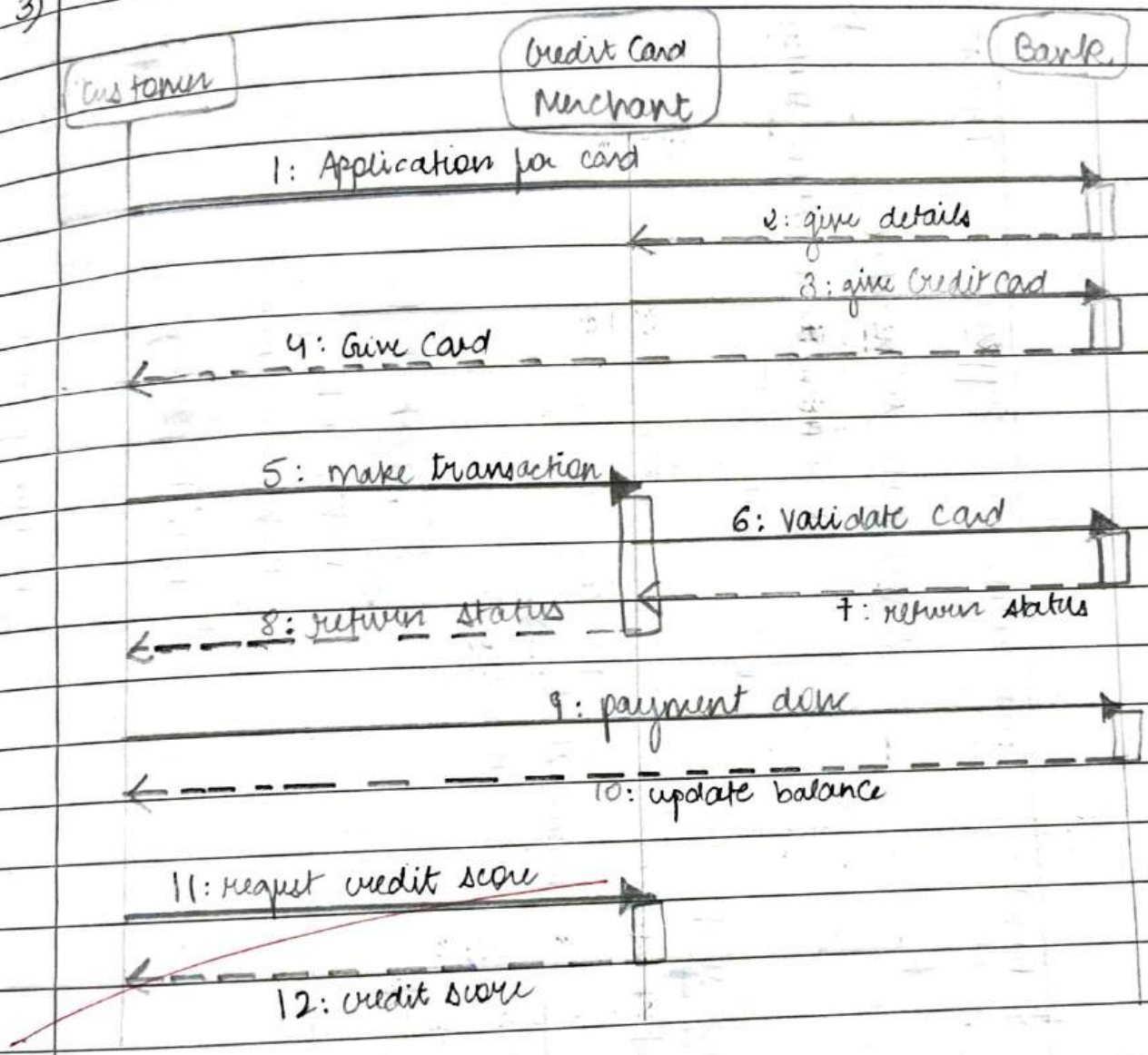
Date __/__/__
Page

2) Library Management System :

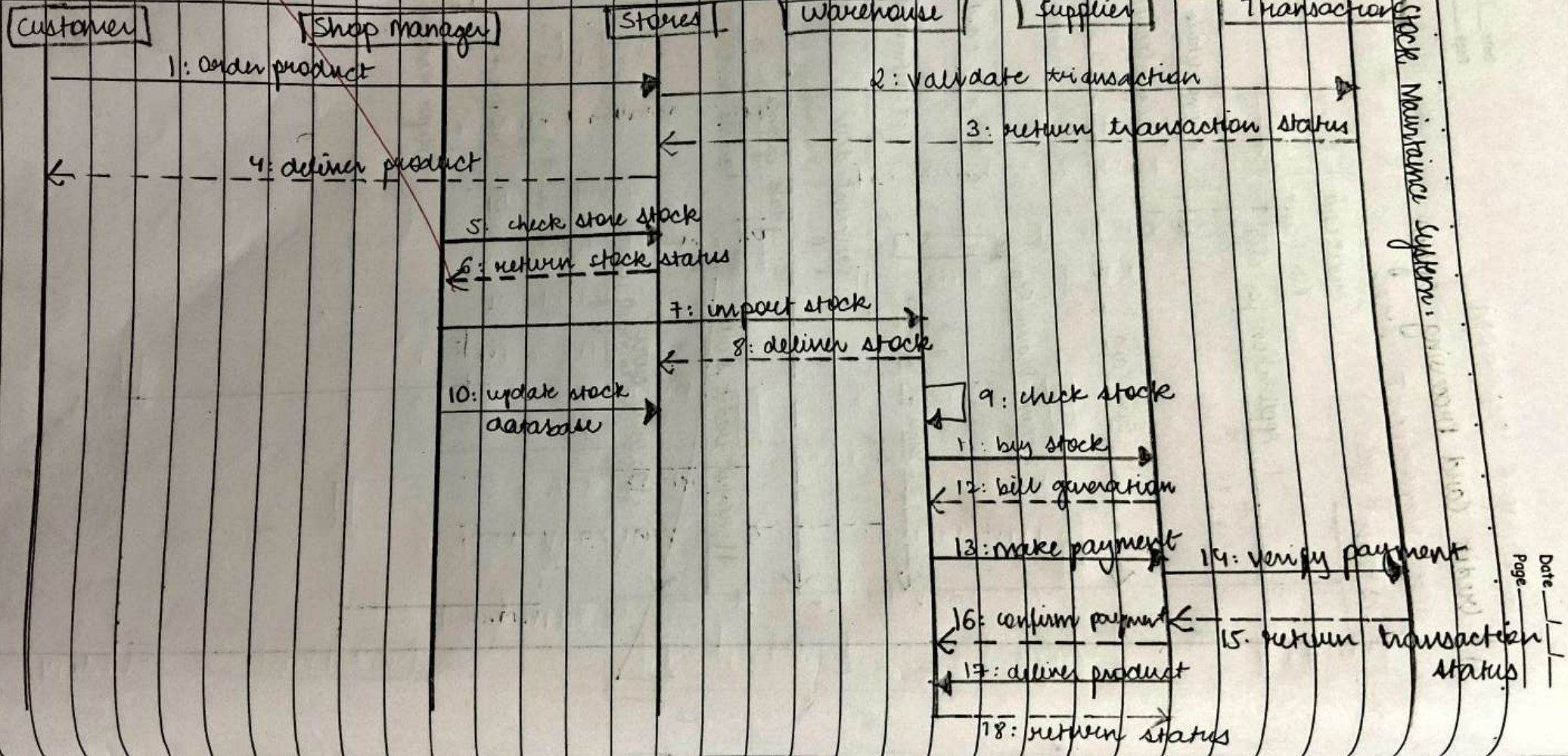


Date __/__/__
Page _____

3) Credit Card Processing System

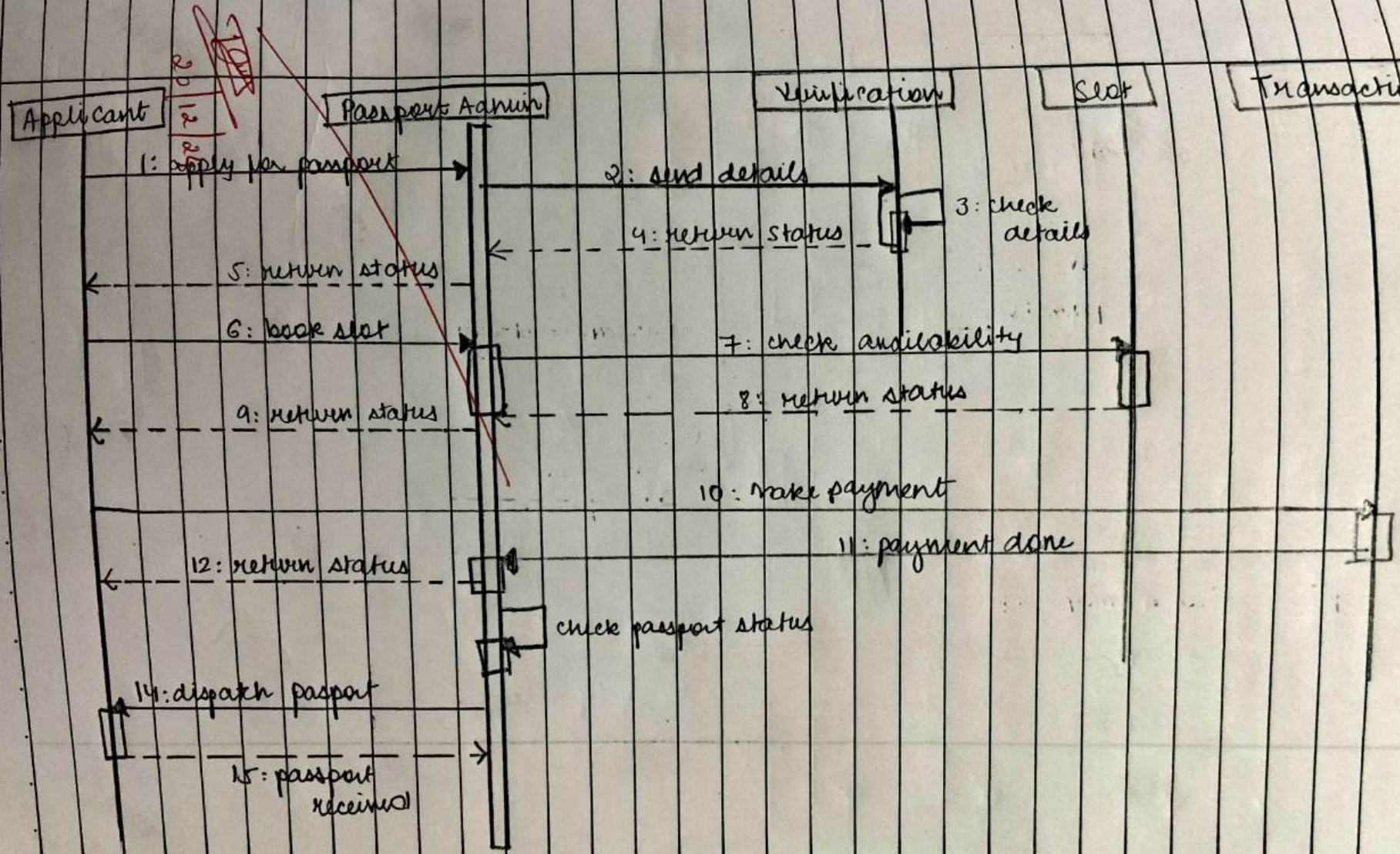


4) Stock Maintaince System:



Date __/__/__
Page

5) Passport Automation System



Date __/__/__
Page