

SRS Document

1. Hotel Management System

1.1 Purpose of this Document

This document outlines the software requirements for the Hotel Management System (HMS).

The purpose is to provide a clear understanding of the system's functionalities, user requirements, and constraints to ensure all stakeholders have a common understanding of what to be developed.

1.2 Scope of this Document

The HMS will facilitate the management of hotel operations, including reservations, check-in/out processes, billing, and reporting. The value to customers includes improved efficiency, enhanced customer service and streamlined operations. Development is estimated to require 6 months with a budget of \$150,000.

Overview

The Hotel Management System will be a web-based application allowing hotel staff to manage bookings, customer information, room availability, and financial transactions. It will include a user-friendly interface for both staff and customers, ensuring a seamless experience.

1. General Description

1.1 User Objectives

- Receptionists: Manage bookings, check-ins, and check-outs
- Management: generate reports and manage overall hotel operations.
- Guests: View and book rooms online

1.2 User characteristics

Staff: Basic computer literacy and familiarity with hotel operations

Guests: Varying levels of technical skills

1.3 Features and Benefits

Online booking: increase customer convenience
Room management: Optimize room availability and pricing

Reporting tools: Helps management make informed decisions

1.4 Importance

The Hotel is crucial for enhancing operational efficiency.

2. Functional Requirements

- User Authentication: users must log in.
- Room Booking: user can search for and book available rooms.
- Check-in/out process: streamlined process
- Billing and Payment processing
- Reporting: revenue reports etc.

3. Interface requirements

User Interface: web based UI, accessible from various devices

APIs: for integrating with 3rd party

Database Interface: secure access to the hotel database

4. Performance requirements

- The system should handle upto 1000 concurrent users.
- Response time for room searches
- Daily backups must occur without system downtime.

5. Design constraints

Must be developed using the latest version of a chosen web framework (e.g. React).
DB must be hosted on a cloud service provider.

6. Non-functional attributes

Security: Data encryption for sensitive information

Availability: accessible on various devices

Scalability: ability to handle increased user load and data.

7. Preliminary Schedule and Budget

Timeline: 6 months

Budget: ₹1,00,000

Development: ₹60,000

Testing: ₹20,000

Deployment and Maintenance: ₹20,000

2. General Description

2.1 User Objectives

- Receptionists: Manage bookings, check-ins, and check-outs.

- Management: generate reports and manage overall hotel operations.

- Guests: View and book rooms online.

2.2 User characteristics

Staff: Basic computer literacy and familiarity with hotel operations.

Guests: Varying levels of technical skill.

2.3 Feature and Benefits

Online Booking: Increase customer convenience.

Room management: Optimize room availability and pricing.

Reporting tools: Helps management make informed decisions.

2.4 Importance

The HIS is crucial for enhancing operational efficiency.

3. Functional Requirements

- User Authentication: users must log in.

- Room Booking: users can search for and book available rooms.

- Check-in/out process: streamlined process.

- Billing and payment processing.

- Reporting: revenue reports etc.

4. Interface requirements

User interface: web-based UI, accessible from various devices.

APIs: for integrating with 3rd party.

Database Interface: secure access to the hotel database.

5. Performance requirements

- The system should handle up to 1500 concurrent users.

- Response time for room searches.

- Daily backups: must occur without system downtime.

6. Design constraints

Must be developed using the latest version of a chosen web framework (e.g. React). DB must be hosted on a cloud service provider.

7. Non-functional attributes

Security: Data encryption for sensitive information.

Availability: accessible on various devices.

Scalability: ability to handle increased user load and data.

8. Preliminary Schedule and Budget

Timeline: 6 months

Budget: \$1,000,000

Development: 1,000,000

Testing: 2,000,000

Development and Maintenance: 2,000,000

20/07/23

Credit Card Processing

Project Description

The credit card processing system is designed to facilitate secure transaction between customers. It must allow users to perform payment, check transaction history and provide gateways.

Problem Statement

With the rise of e-commerce and online payments, there is a need for a secure and efficient system to process credit card payments and ensure fraud detection.

System Requirements

Functional requirements: user authentication, credit card validation, transaction history and reporting.

Non-functional requirements: high security, credit card data, minimum downtime, fast resource times, scalability.

Domain Requirements: Integration with bank system, supports different currencies, able to financial standards.

Design constraints

- strong encryption protocols
- limited choice of APIs
- database constraints due to high transaction volume
- limited choice of libraries for fraud detection
- limited choice of protocols

Preliminary schedule and budget

Preliminary schedule:

requirement analysis	2 weeks
system design	4 weeks
development	10 weeks
testing	3 weeks
deployment	1 week
Total duration	20 weeks

Preliminary Budget

Development team	\$40k
Hardware	\$5k
Software license	\$10k
Miscellaneous costs	\$5k
Total Budget	\$80k

Cost
of

Library Management System (LMS)

1. Introduction

1.1 Purpose of this Document:

This document outlines the requirements for developing a Library Management System. The purpose is to provide a system that simplifies library operations, enhance user experience, and improves overall efficiency.

1.2 Scope of this Document:

This document sets the design, development, and implementation of an LMS. It will track book inventory, manage user memberships, and allow for borrowing and returning of books. The system will be designed to minimize manual labor and reduce human error.

1.3 Overall General Description:

The LMS will allow library staff and patrons. Users can search for books, issue or borrow them, and check their account details. Librarians can manage inventory and track borrowing history. The system aims to improve the efficiency management while reducing errors.

1.4 Overview

The LMS will allow librarians to manage

book collections, handle user data, and track borrowing activities. It will include modules for search, reservation, issuing, and returning of books, along with late fee calculation.

2. Functional Requirements

User authentication for members and staff.

Book catalog management (add, remove, update books).

Issue and return tracking with fine calculation for overdue books.

Search and reservation functionality for users.

Generate reports on book inventory, borrowing history, and fines.

3. Interface Requirements

Web-based interface for users and staff.
Integration with external databases for catalog updates.

Notifications for overdue books through email or SMS.

4. Performance Requirements

Response time should be under 2 seconds for common queries.

System should handle up to 100 concurrent users.

Database storage capacity for 1 million book records.

2. Design Constraint

Must use SQL database for storage compatible with Linux, windows

Responsive UI for both desktop and mobile devices

3. Non-Functional Attributes

- High reliability with 99.9% uptime.
- Ability to handle growing user base and track inventory.
- Secure user authentication and data encryption.

4. Preliminary Budget and Schedules

\$30,000

Development: 5 months

Testing: 4 weeks

Deployment: 1 week

Maintenance: 1 week

Minor bugfixes: 1 week

Stock Maintenance System (SMS)

1. Introduction

1.1 Purpose of this Document

This document specifies the requirements for a stock maintenance system, aimed at automating the process of managing stock levels in a business.

1.2 Scope of this Document

The SMS will be designed to track inventory levels, manage orders, alert for restocking, minimize manual inventory tracking and ensure stock levels are optimal.

1.3 Overview

The system will automate stock tracking for warehouses or retail environments, ensuring that inventory levels are maintained. It will provide real-time stock update reports.

2. General Description

This system is targeted at inventory managers and warehouse staff. Users can check current stock levels, place orders for reordering, generate reports.

3. Functional requirements

- Stock tracking multiple items with real-time updates
- Order alerts when stock level fall below predefined thresholds
- User management for inventory staff and administrators
- Report generation
- Integration with accounting and sales systems

4. Interface requirements

- Web-based interface for stock managers
- API integration with external sales platforms.
- Real-time notifications through email or system alerts.

5. Performance requirements

- System response should be within 1 second for stock queries
- Handle up to 50 concurrent users and 5,00,000 stock items.
- Daily processing of up to 10,000 transactions.

6. Design Constraints

- System must use MySQL or PostgreSQL database
- Compatible with both cloud and on-premise servers
- Must provide a mobile version for easy access on the go

7. Non-functional attributes

- Must ensure data integrity and backup regularly.
- High availability with a minimum of 99.8% uptime.
- Secure access control for different user roles.

8. Preliminary Schedule and Budget

- System will be developed in 3 months with an approximate budget of \$15,00,000 involving development, integration and testing.

Development	\$6,00,000
Testing	\$4,00,000
Deployment	\$1,00,000
Maintenance	\$50,000
Miscellaneous	\$50,000

3 Passport automation system

1. Introduction

1.1 Purpose of this Document

This document outlines the design and requirements for a passport automation system aimed at simplifying and speeding up the passport application process.

1.2 Scope of this Document

The PAS will automate passport application processing, verification, and status tracking. The system will reduce paperwork, improve the speed of processing, and provide users with real-time status updates.

1.3 Overview

The system will allow users to apply for passports online, track their application status, and schedule appointments for verification. Officials will be able to verify applications, update statuses, approve or reject applications.

2 General Description

The system is intended for both passport applicants and government officials. Applicants will submit their documents, pay fees, track application status. Government officials will verify documents, conduct background checks,

and approve applications. The system ensures transparency and reduces processing time.

2 Functional Requirements

- User registration and login for passport applicants
- Document submission and fee payment for passport application
- Status tracking and appointment scheduling
- Application verification and approval
- Automated email/SMS notifications for application status

3 Interface Requirements

- Web-based interface for users and government officials
- Secure communication with government databases for background checks
- Integration with payment gateways for fee submission

5 Performance Requirements

System should process up to 100,000 applications per month.

Must handle up to 200 concurrent users during peak times.

Status update within 1 second after change is made.

08/10/24

6. Design constraints

- must comply with government security standards
- system should run on a government approved cloud platform.
- use of Java and SSL for backend and database management.

7. Non-functional attributes

- High level of security for sensitive personal data
- System must be portable and adaptable to different regions
- Scalability to handle increasing numbers of applications.

8. Preliminary Schedule and Budget

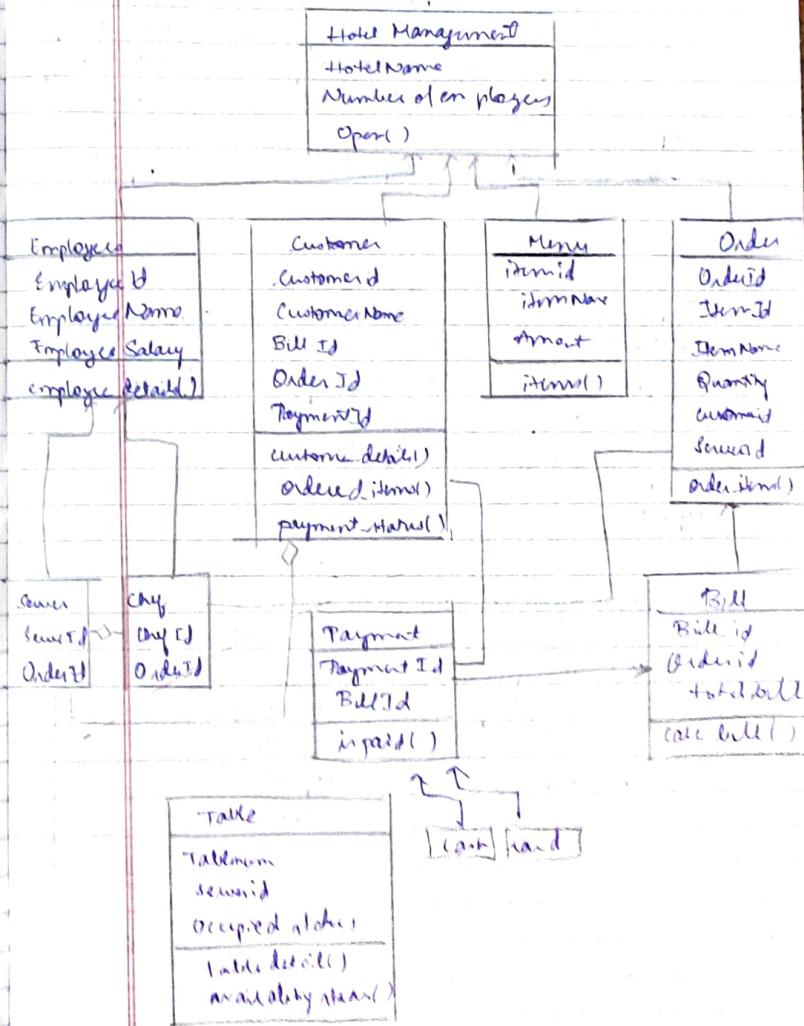
The project will take approximately 6 months to complete, with an estimated budget of ₹ 1,60,00,000.

Development	80,00,000
Testing	60,00,000
Deployment	10,000
Minor Enhancements	30,000
Maintenance	50,000

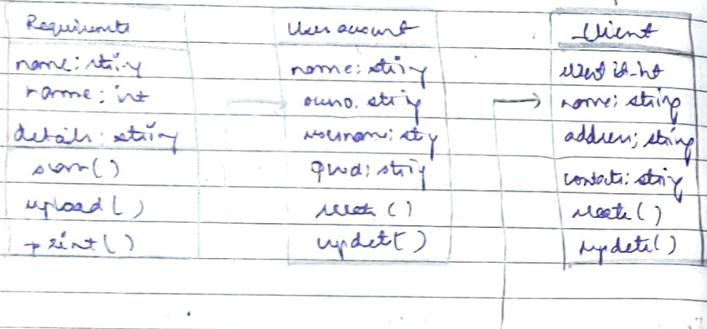
20/10/24

Plan Diagram

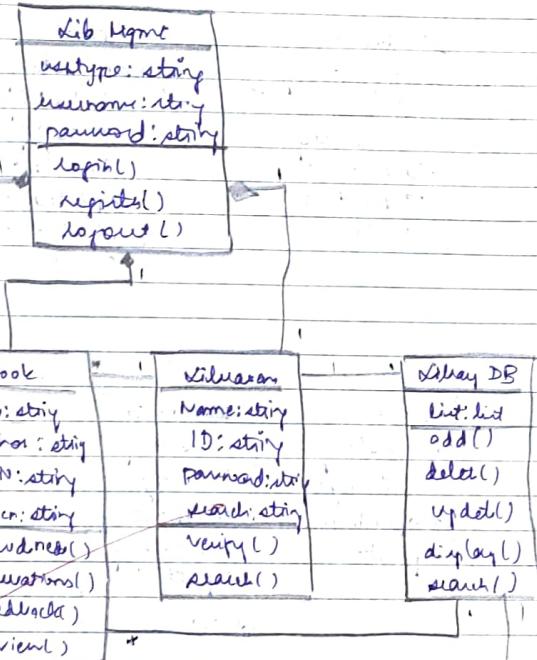
Hotel Management Systems



Credit card processing system.



Library management system.



Employee

userId: int
username: string
password: string
add()
update()

Admin

userId: int
price: string
reset()
update()

Credit Card

cardId: int
owner: int
date: int
status: int
creat()
update()
terminate()

User

name: string
Id: string
verify()
checkPwd()
getInfo()

Book

title: string
author: string
ISBN: string
Publisher: string
showDetails()
Illustrations()

Library

Name: string
ID: string
password: string
search: string
Verify()
search()

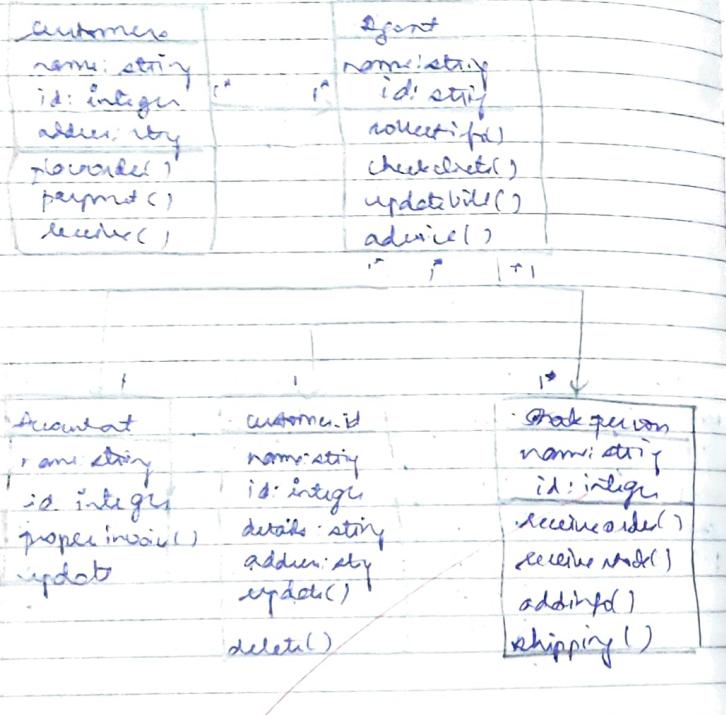
Library DB

list: list
add()
delete()
update()
display()
search()

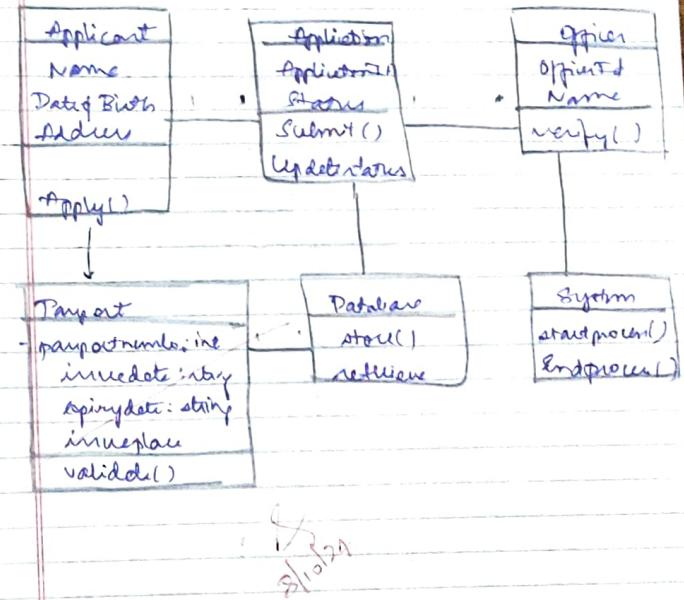
Account

unborrowed
borrowed
notreturned
return
deflate()

Stock Maintenance System



Transport Management System



State Diagrams

Hotel Management System

Start

Reservation

search

check availability

Not available

available

Make reservation

↓ here

Booking Reservation

↓ process

confirm reservation

↓ success

checked in

checked out

pay bill

payment

exit

Credit card processing

start

Idle

↓ card inserted

validation

↓ card read successfully

ready to

↓ card in inserted

choosing transaction

↓ transaction chosen

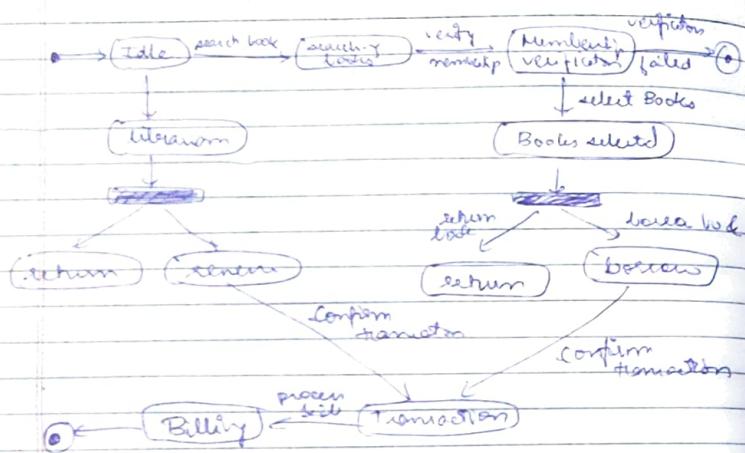
Performing Transaction

↓ finished transaction

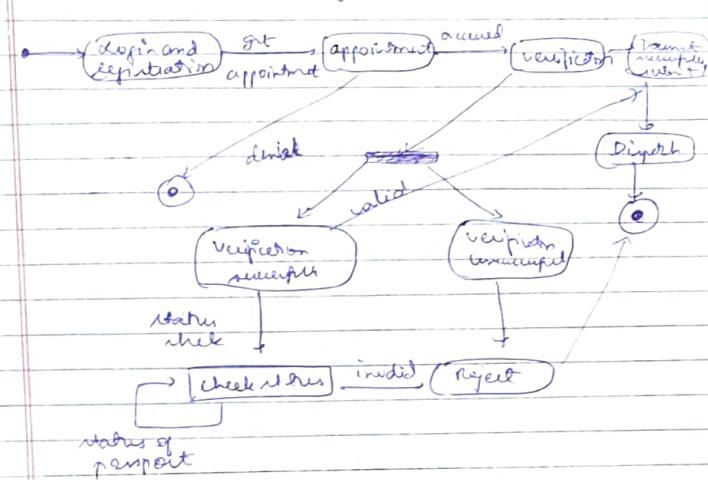
Exit End

Card read

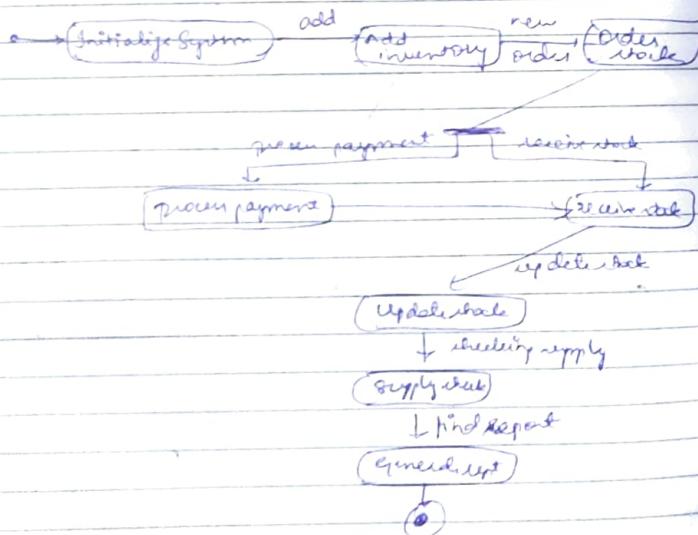
3. Library Management System



4. Panpet automation System



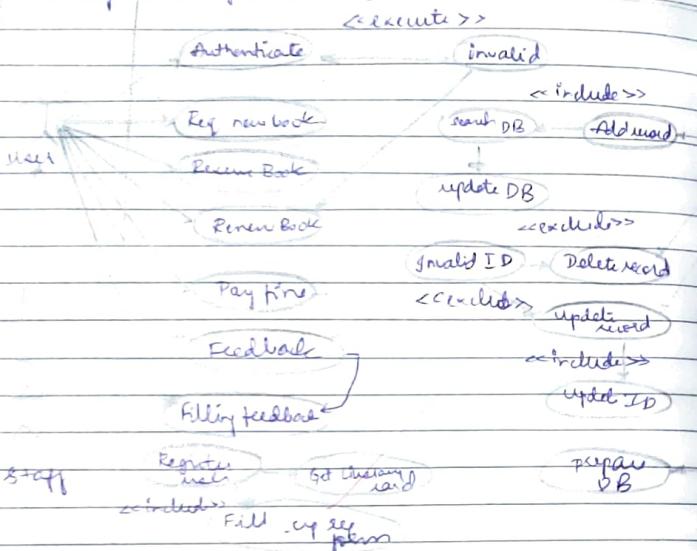
5. Stock maintenance system



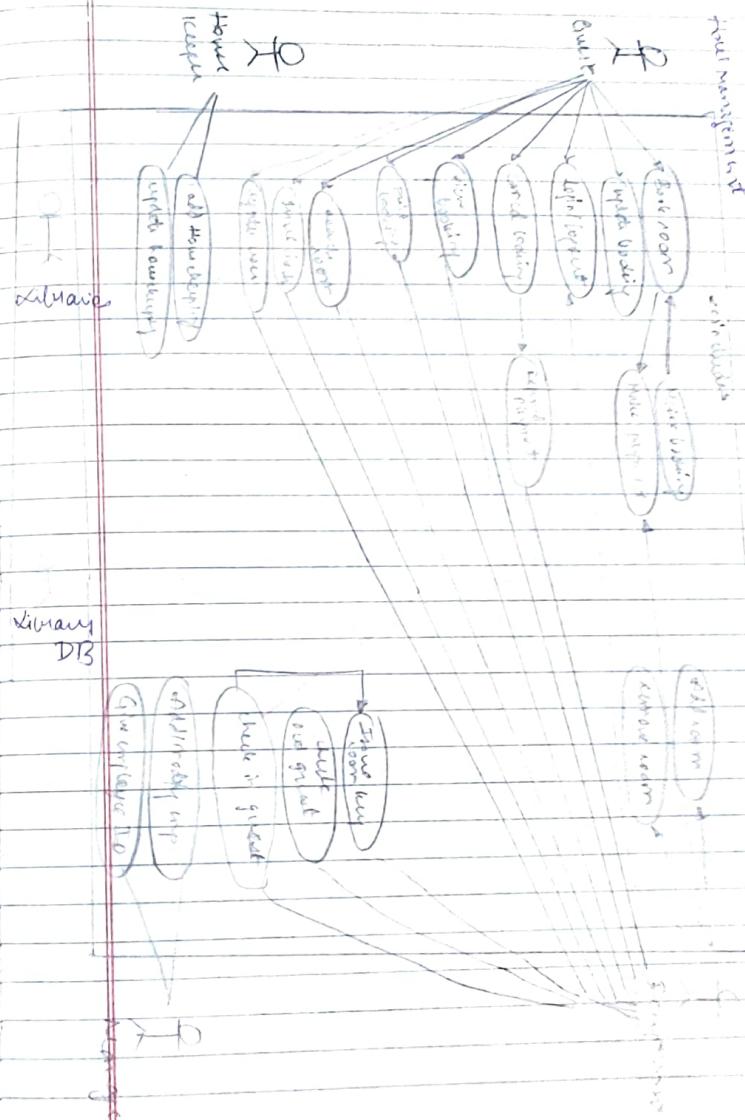
LAB-03

1 USE CASE DIAGRAMS

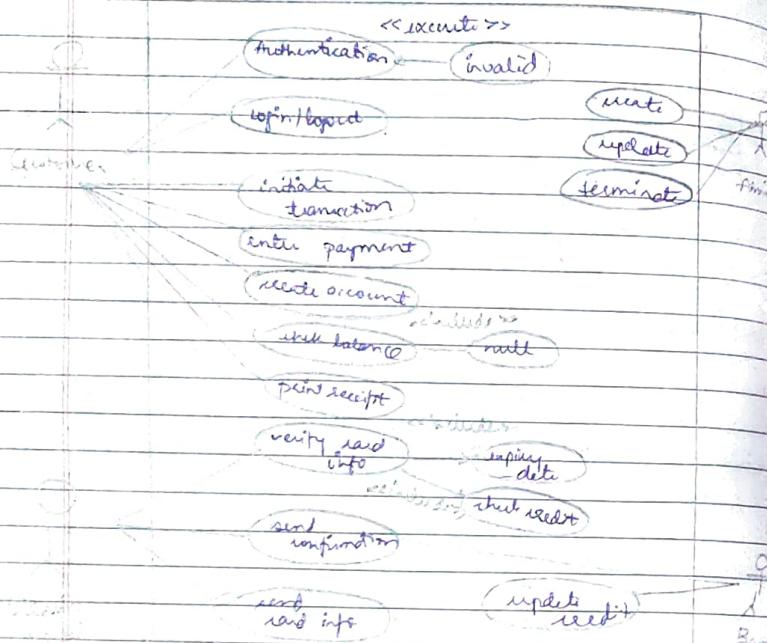
i) Library Management System



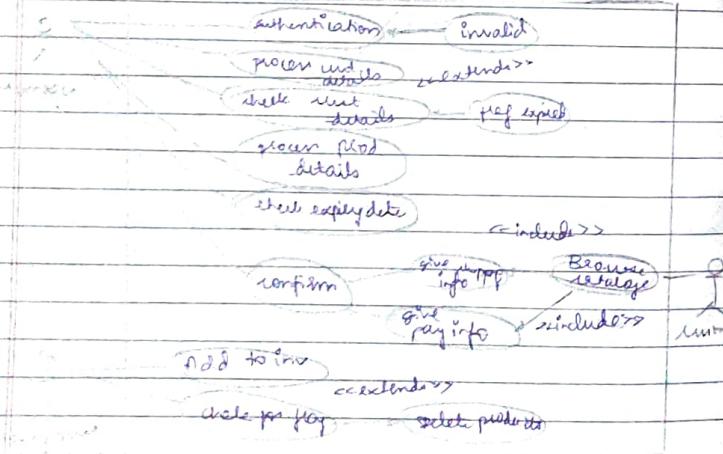
ii) Hotel Management System



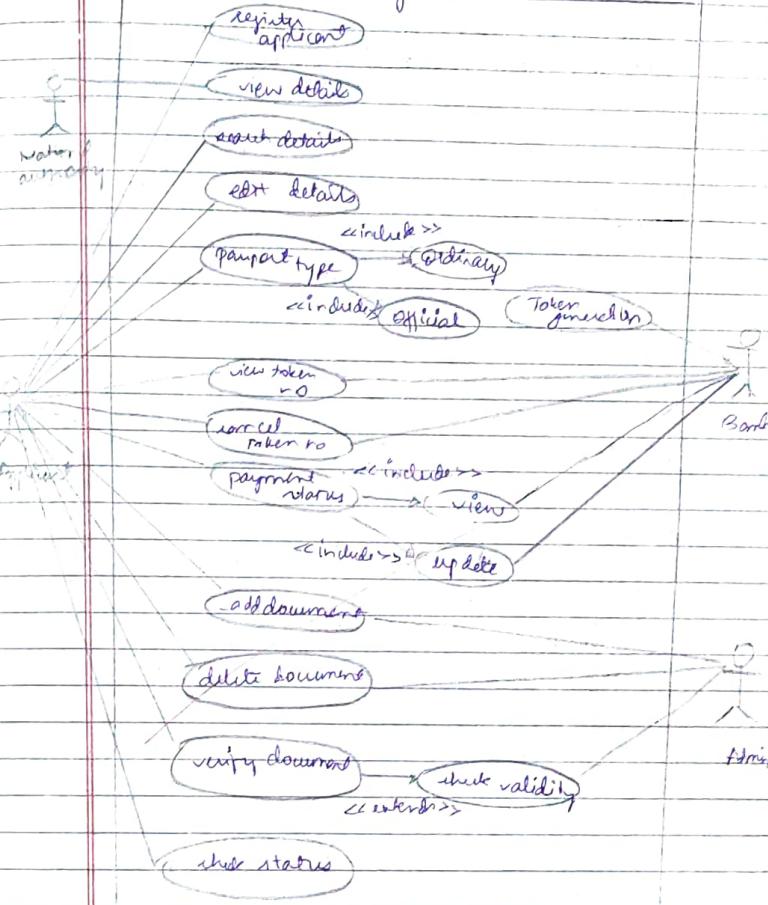
2. Credit Card Processing System



3. Stock Maintenance System: executors

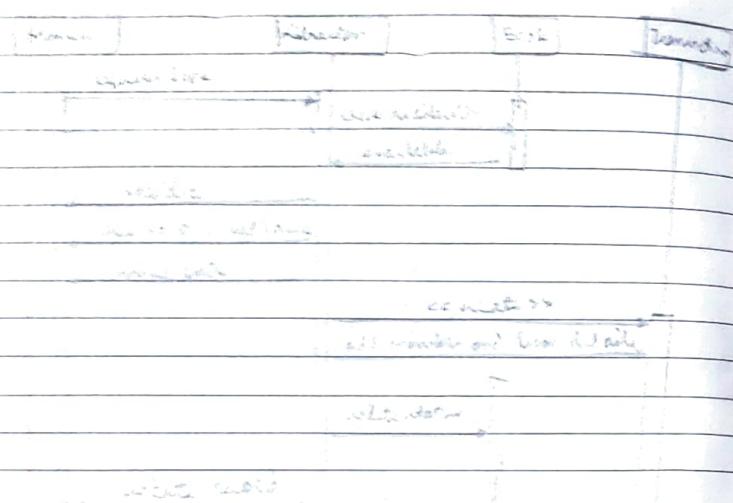


5. Passport Automation System

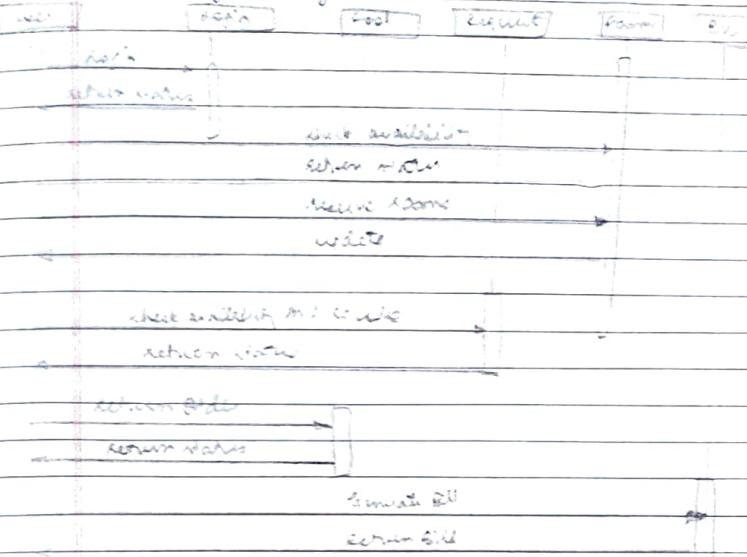


Sequence Diagram

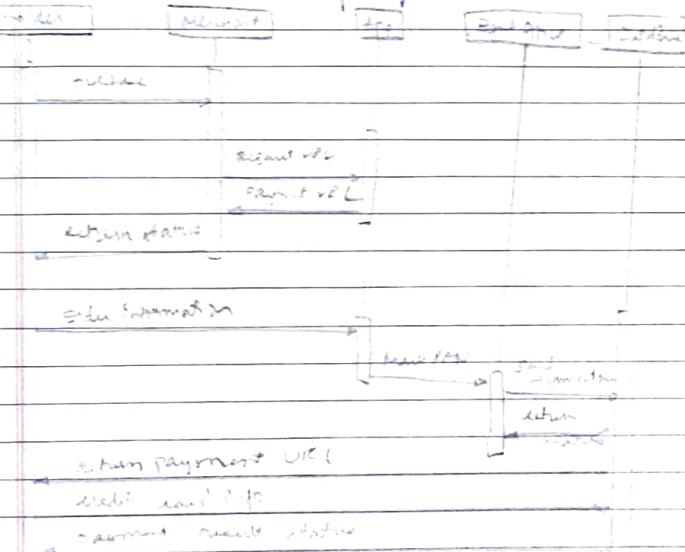
Library Management System



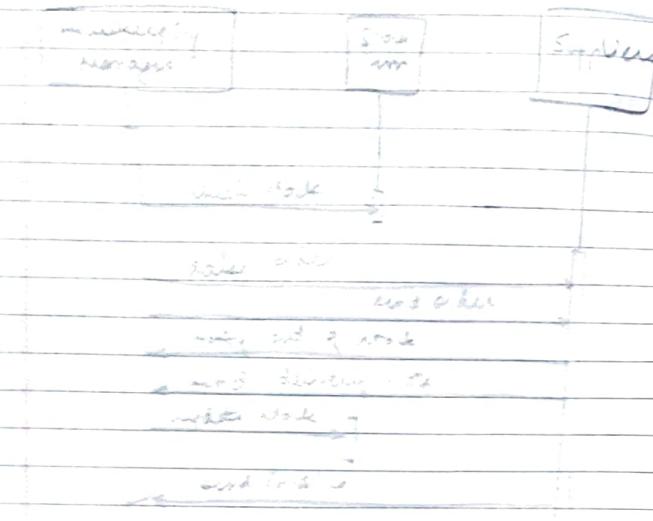
Hotel Management System



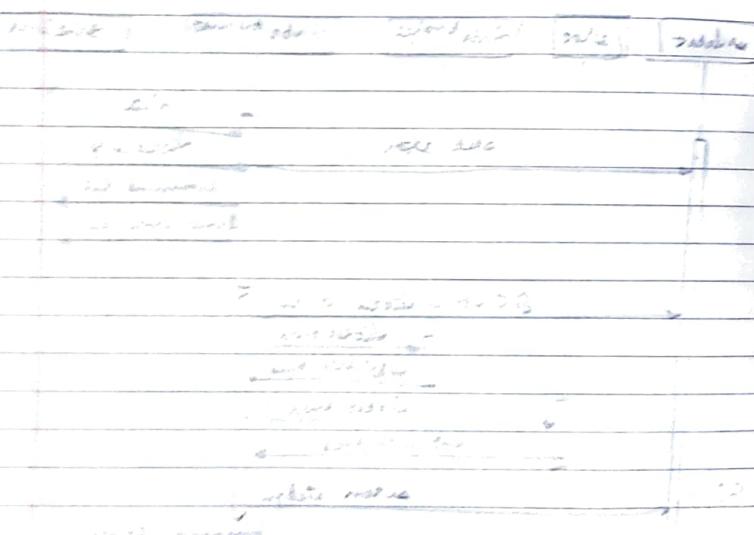
Credit card processing system



2) Dark Maintenance System

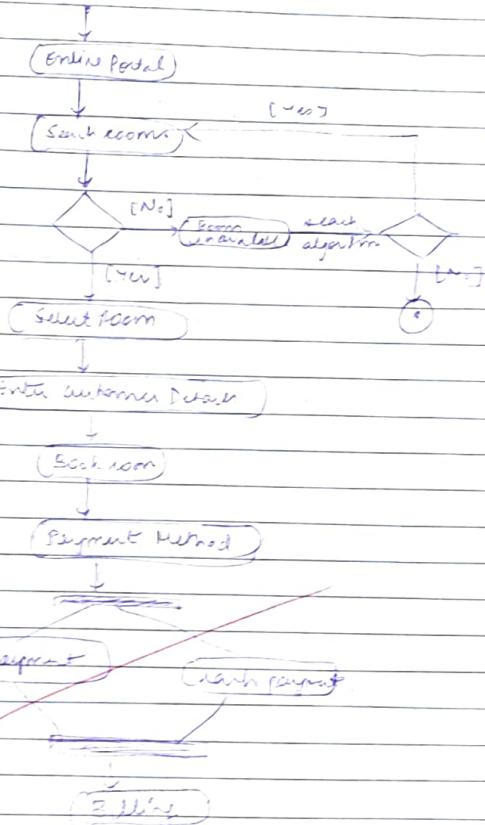


• Fastest Autonode system



Activity Diagram

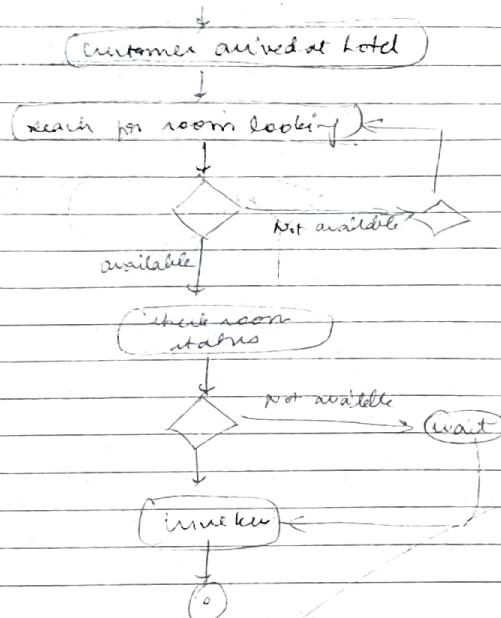
i) Hotel Management System



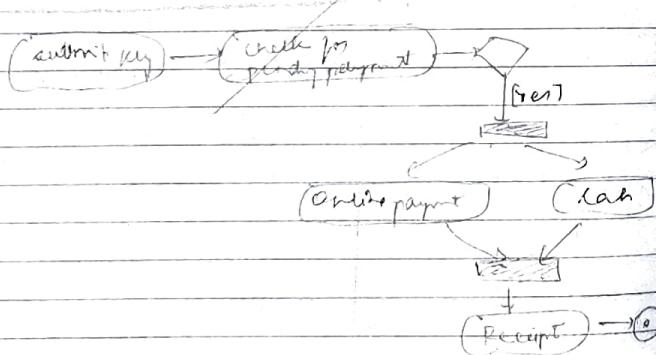
Activity Diagram

1) Hotel Management

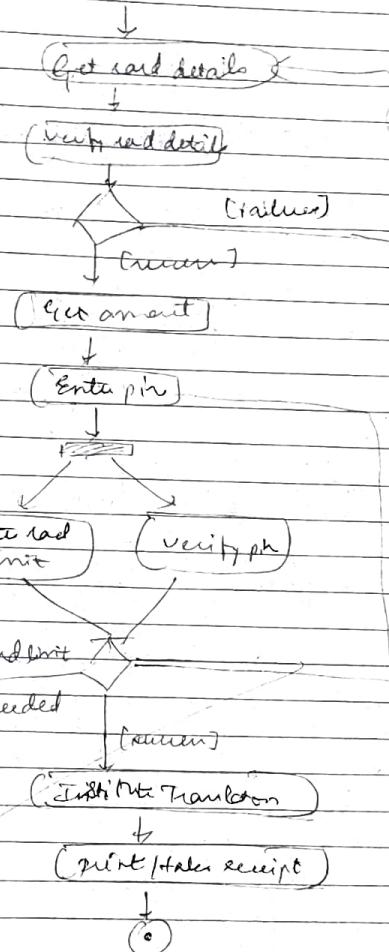
Check In



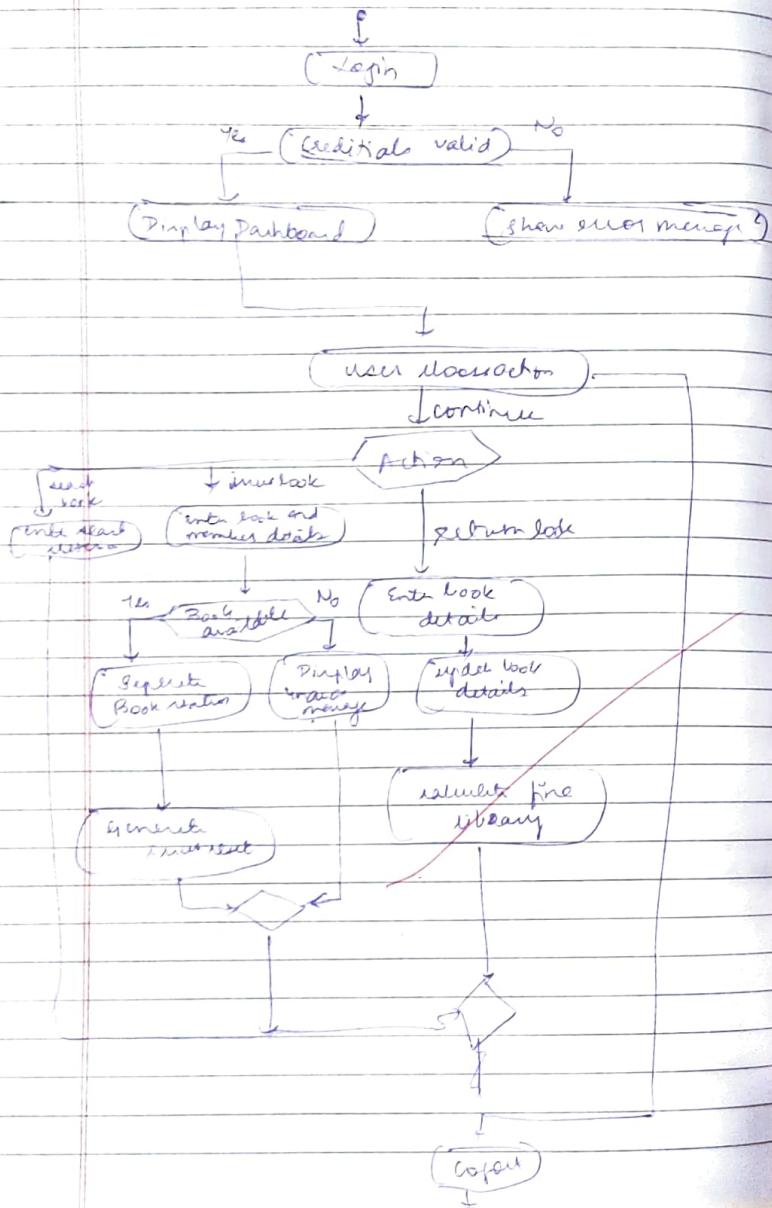
Check Out



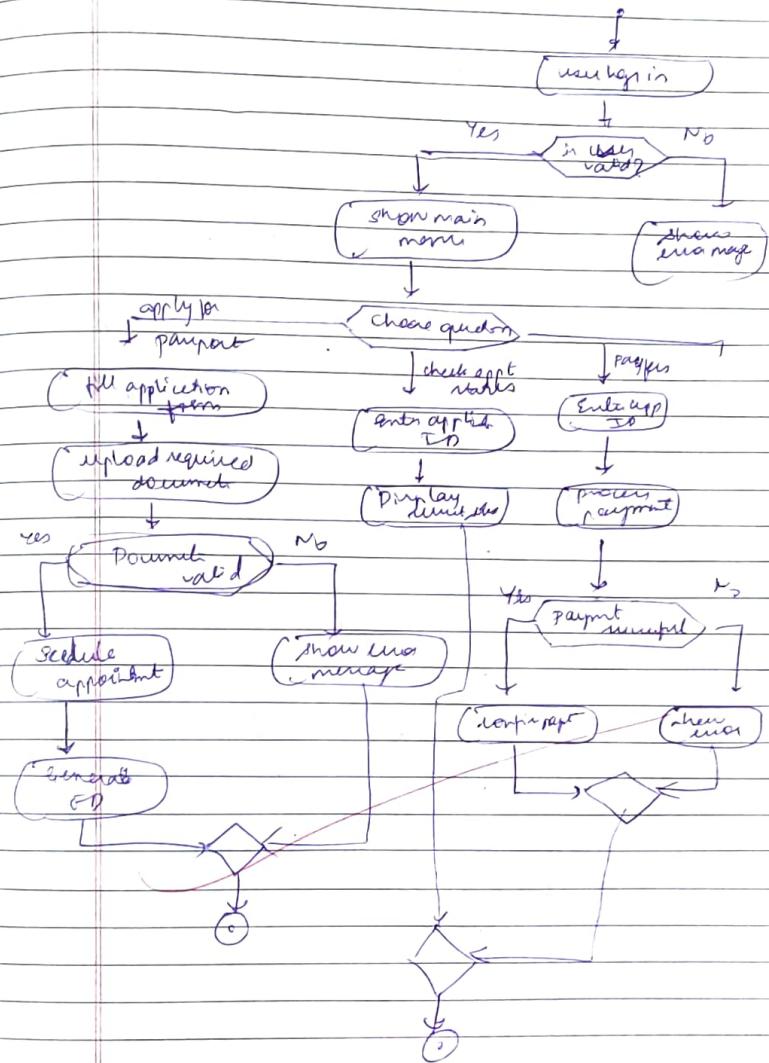
2) Credit card processing:



3) Library Management



4) Passport automation system



5) Stock maintenance system

