VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED MODELLING

Submitted by

Mogaveera Sahil Shekar
(1BM22CS155)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
September-2024 to January-2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED MODELING**" was carried out by **Mogaveera Sahil Shekar (1BM22CS155),** who is a Bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-2025. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Modelling- (23CS5PCOOM)** work prescribed for the said degree.

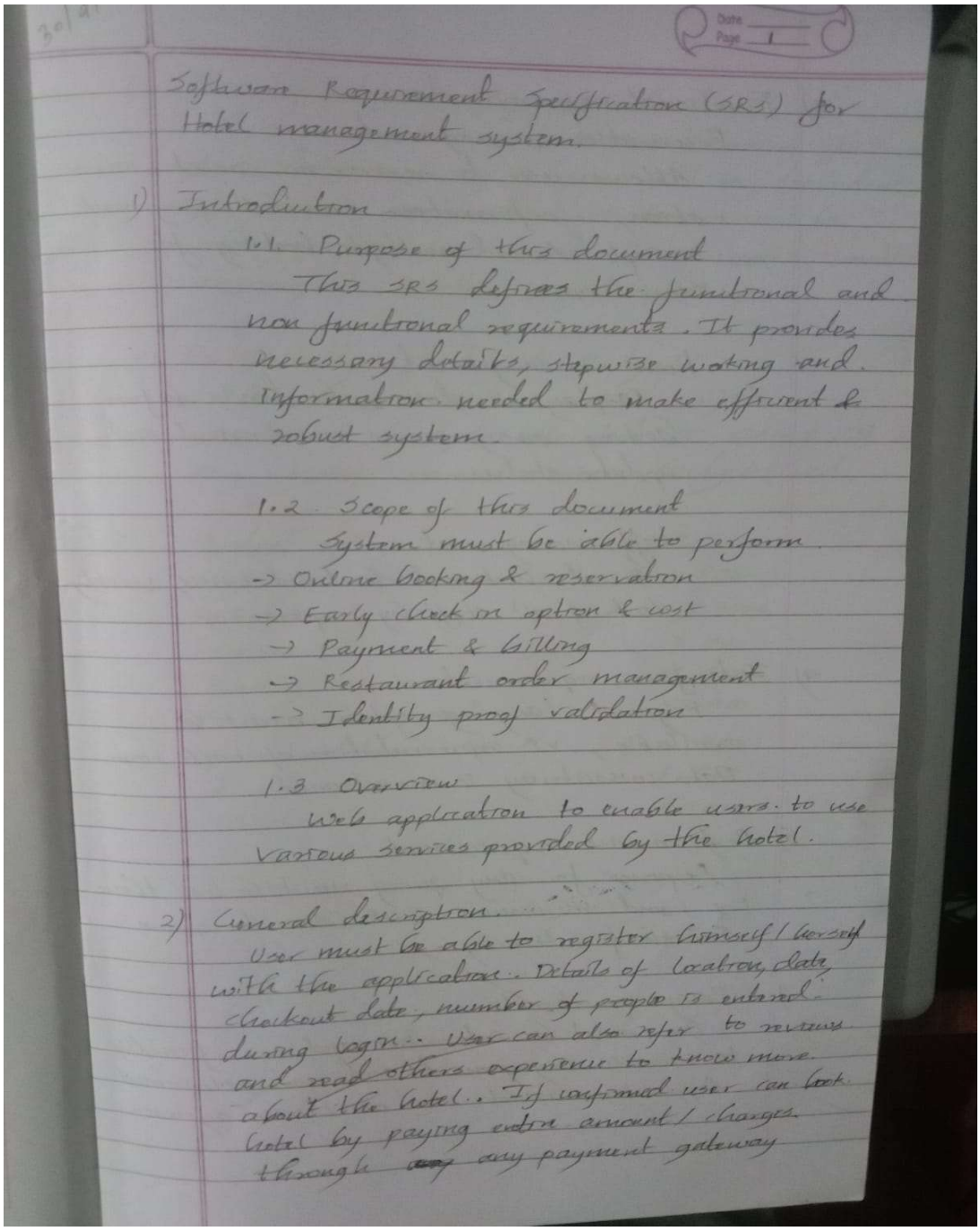
Dr. Latha N.R.
Associate Professor
Department of CSE
BMSCE, Bengaluru

Dr. KAVITA SOODA
Professor and Head
Department of CSE
BMSCE, Bengaluru

Table of Content

Hotel Management System

a. SRS Document



Software Requirement Specification (SRS) for Hotel management system.

1) Introduction

1.1 Purpose of this document

This SRS defines the functional and non functional requirements. It provides necessary details, stepwise working and information needed to make efficient & robust system.

1.2 Scope of this document

System must be able to perform.

-> Online booking & reservation
-> Early check in option & cost
-> Payment & billing
-> Restaurant order management
-> Identity proof validation

1.3 Overview

Web application to enable users to use various services provided by the hotel.

2) General description.

User must be able to register himself / herself with the application. Details of location, date, checkout date, number of people is entered during login. User can also refer to menus and read others experience to know more about the hotel. If confirmed user can book hotel by paying extra amount / charges through ~~any~~ any payment gateway

3) Functional requirements

Reservation management

Allows user to reserve or cancel reservation, confirmation email & payment receipt. Avoid multiple booking for same room

Room management

Display all do's and don'ts before booking assign rooms on arrival and update status

Staff management.

Record check in & check out times, weekly check of all staff by supervisor.

4) Interface requirement

Regional language option must be available, VR representation of each room Data consistency is maintained

5) Performance requirement

Response for any query must be less than 1 sec and minimum 8GB RAM to ensure smooth functioning.

6) Design constraints

UI must be interactive and user friendly to increase engagement and user retainment Application cannot run on LINUX systems

7) Non functional requirements.

System must use RSA 2048 hashing for ~~dat~~ data security & must not be dependent ~~on~~ on one architecture. System must be able to process & handle 10000 users simultaneously if needed must scale up.

8) Preliminary schedule & Budget.

Project must be completed within 6 months & budget allocated is $9600. If any changes made reserve budget used.

Budget split

Software development	$3840
Hardware	$1920
Licenses	$960
Quality assurance	$960
Project management	$480
Documentation	$480
Maintenance	$960
Total	$9600

a. Advanced Class Diagram:

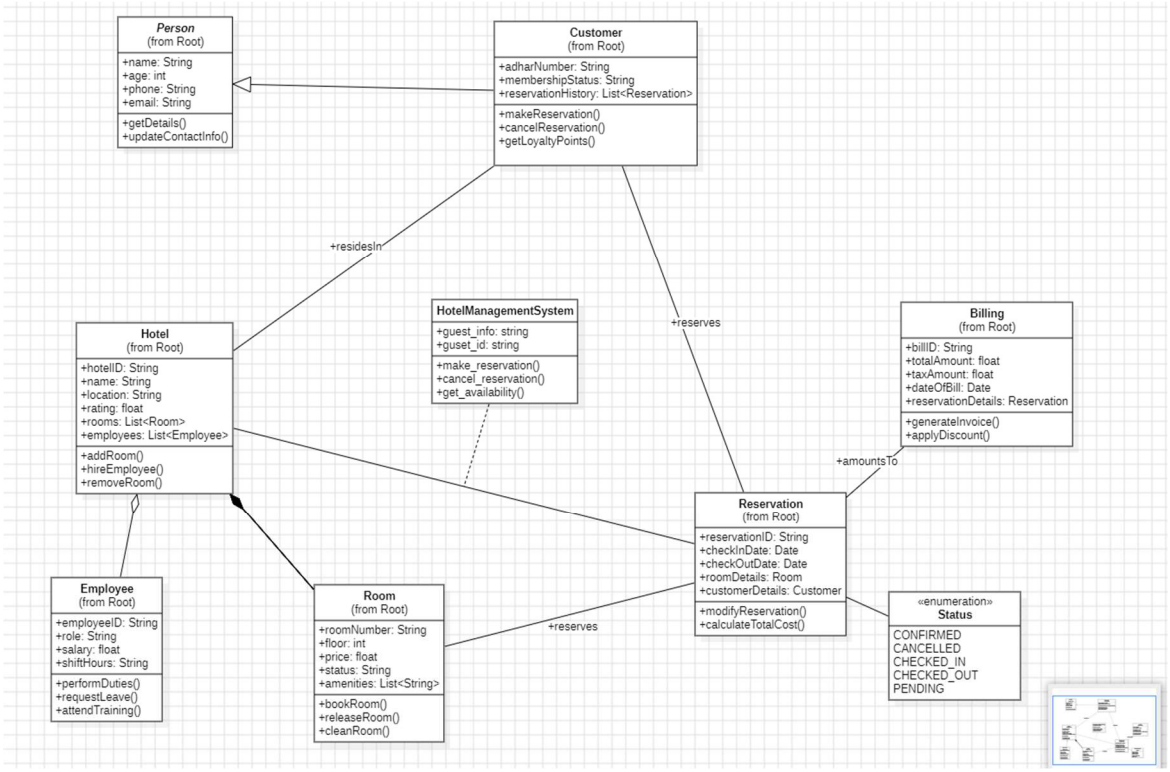
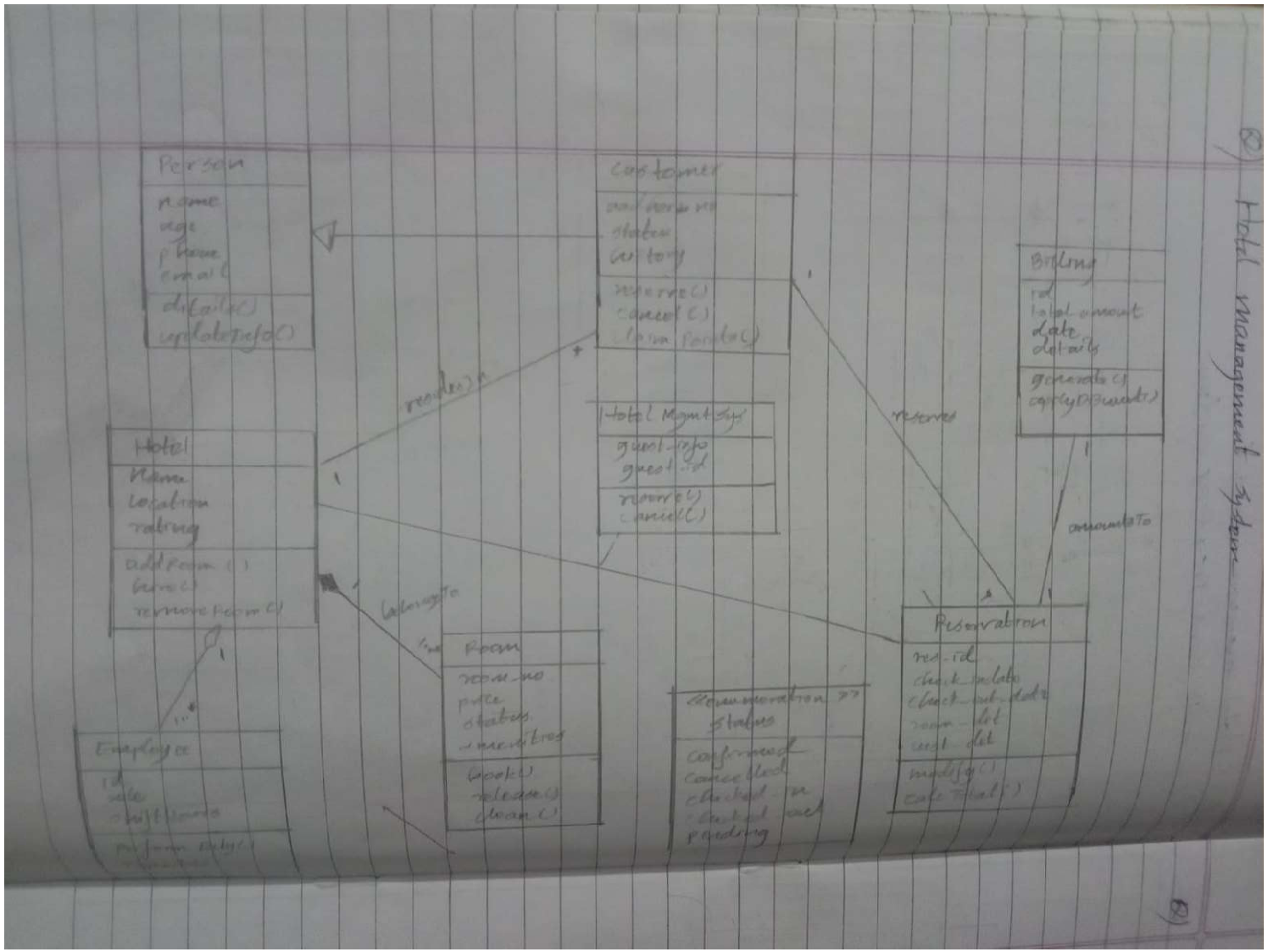


Fig 1.2 : Advanced Class Diagram for hotel management system

- **Person and Customer**: The Person class serves as a base for Customer, which includes details like membership and reservation history. Customers can make and cancel reservations.
- **Hotel**: The central entity managing Rooms and Employees. It includes methods to add or remove rooms and hire staff.
- **Room**: Represents individual rooms in the hotel, with details like room number, price, status, and amenities. Rooms can be booked, released, or cleaned.
- **Employee**: Represents hotel staff, with details like role, salary, and methods to perform duties or request leave.
- **Reservation and Billing**: The Reservation class handles booking details like check-in/check-out dates, room, and customer information. The Billing class generates invoices and applies discounts for reservations.
- **Hotel Management System**: Manages reservations, cancellations, and room availability, acting as the core system.


Hotel Management System
Person
name
age
phone
email()
display()
updateInfo()
Customer
status
history
reserve()
cancel()
payBill()
Billing
id
total_amount
date
details
generate()
applyDiscount()
Hotel Mgmt Sys
guest_info
guest_id
reserve()
cancel()
Hotel
Name
location
rating
addRoom()
hire()
removeRoom()
reserves
belongsTo
amountTo
Room
room_no
price
status
amenities
book()
release()
clean()
Reservation
res_id
check_in_date
check_out_date
room_list
cust_det
modify()
cancel()
Reservation Status
Confirmed
Cancelled
Checked_in
Checked_out
Pending
Employee
id
role
shift
performDuty()

b. Advanced State Diagram:

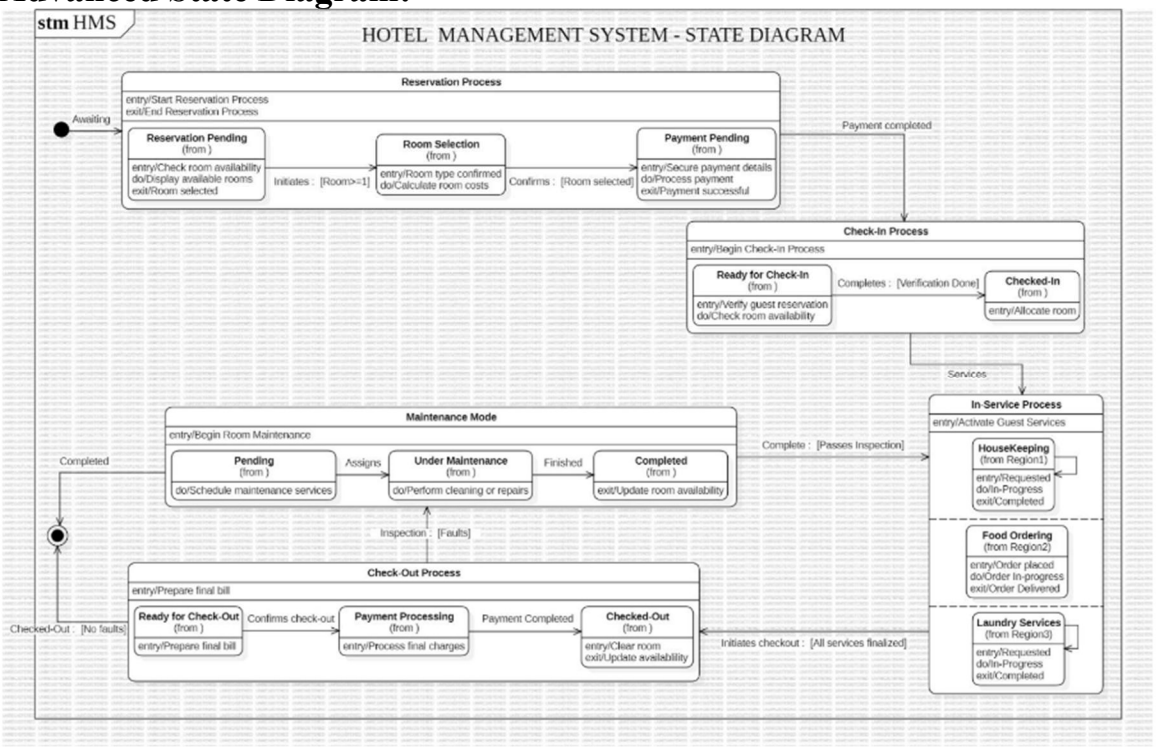
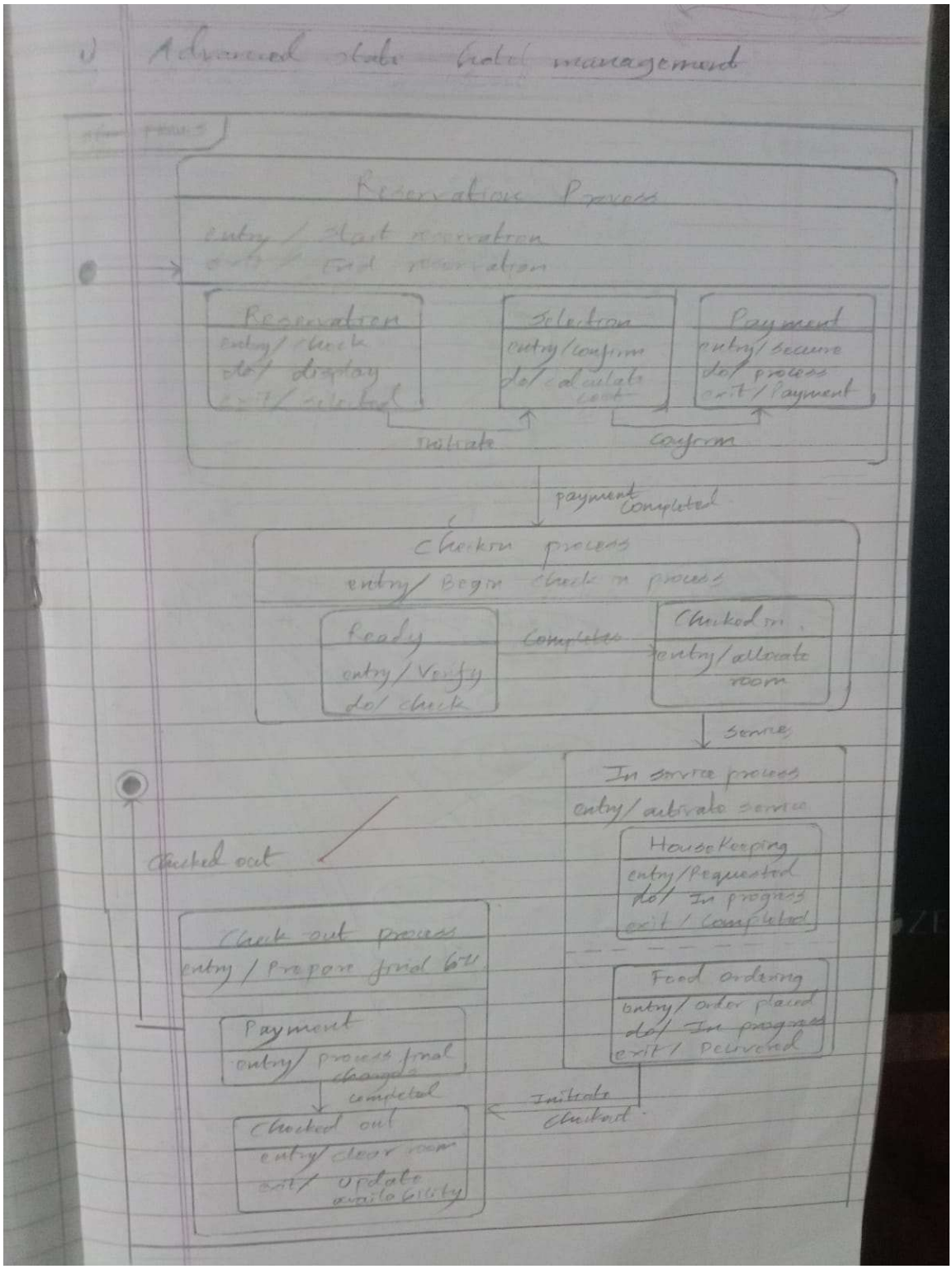


Fig 1.3


Advanced state based management
Reservation Process
entry / start reservation
exit / end reservation
Reservation
entry / check
do / display
exit / select
Selection
entry / confirm
do / calculate cost
Payment
entry / secure
do / process
exit / Payment
Validate
Confirm
payment Completed
Checkin process
entry / Begin check in process
Ready
entry / Verify
do / check
Completes
Checked in
entry / allocate room
Service
In service process
entry / activate service
Housekeeping
entry / Requested
do / In progress
exit / Completed
Food ordering
entry / order placed
do / In progress
exit / Delivered
Checked out
Check out process
entry / Prepare final bill
Payment
entry / process final charges
completed
Checked out
entry / clear room
exit / Update availability
Initiate Checkout

c. Use Case Diagram

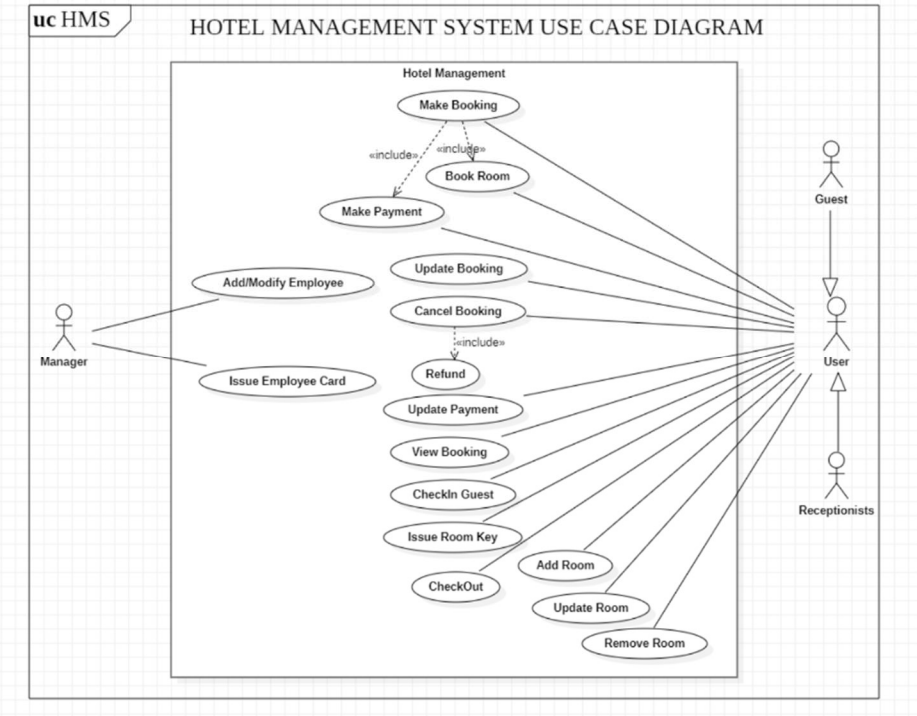
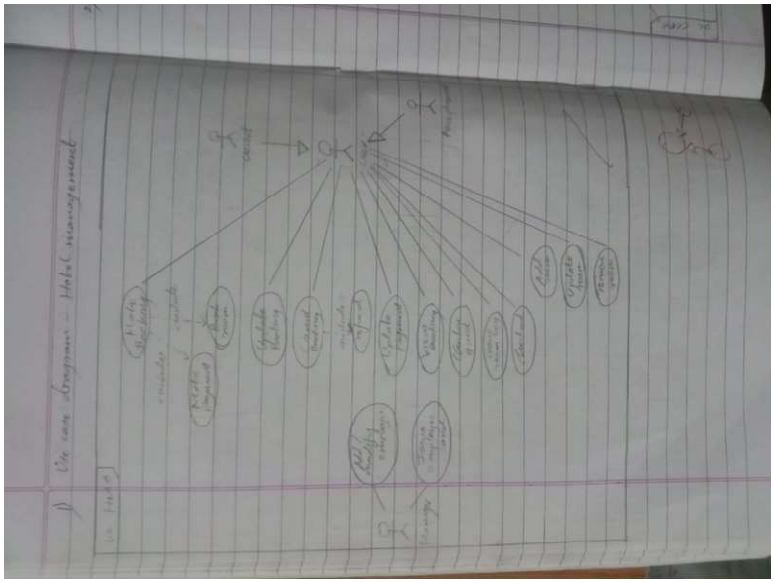


Fig 1.4



d. Sequence Diagram

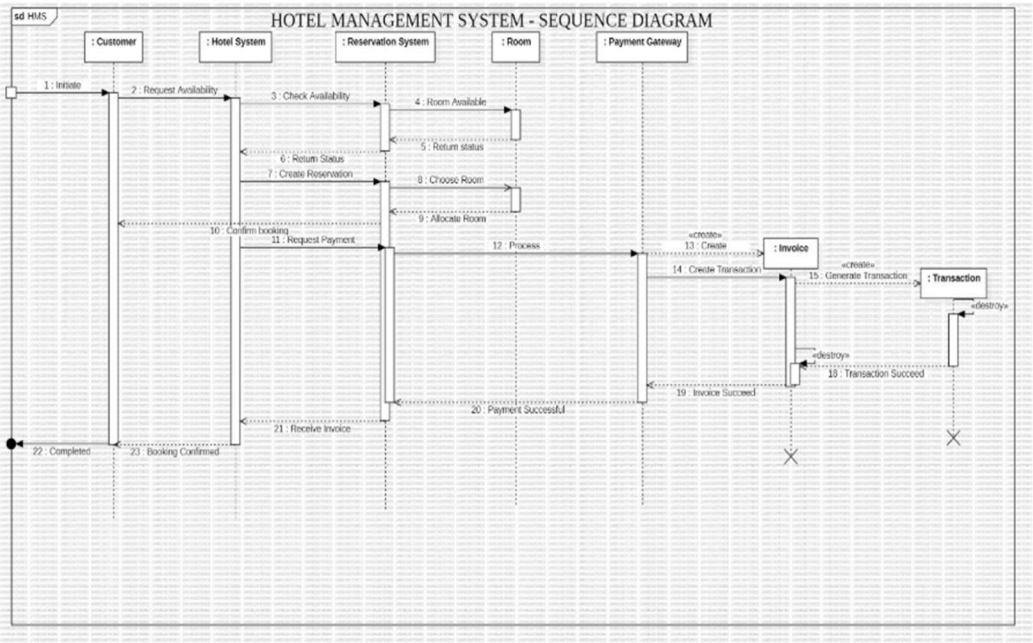
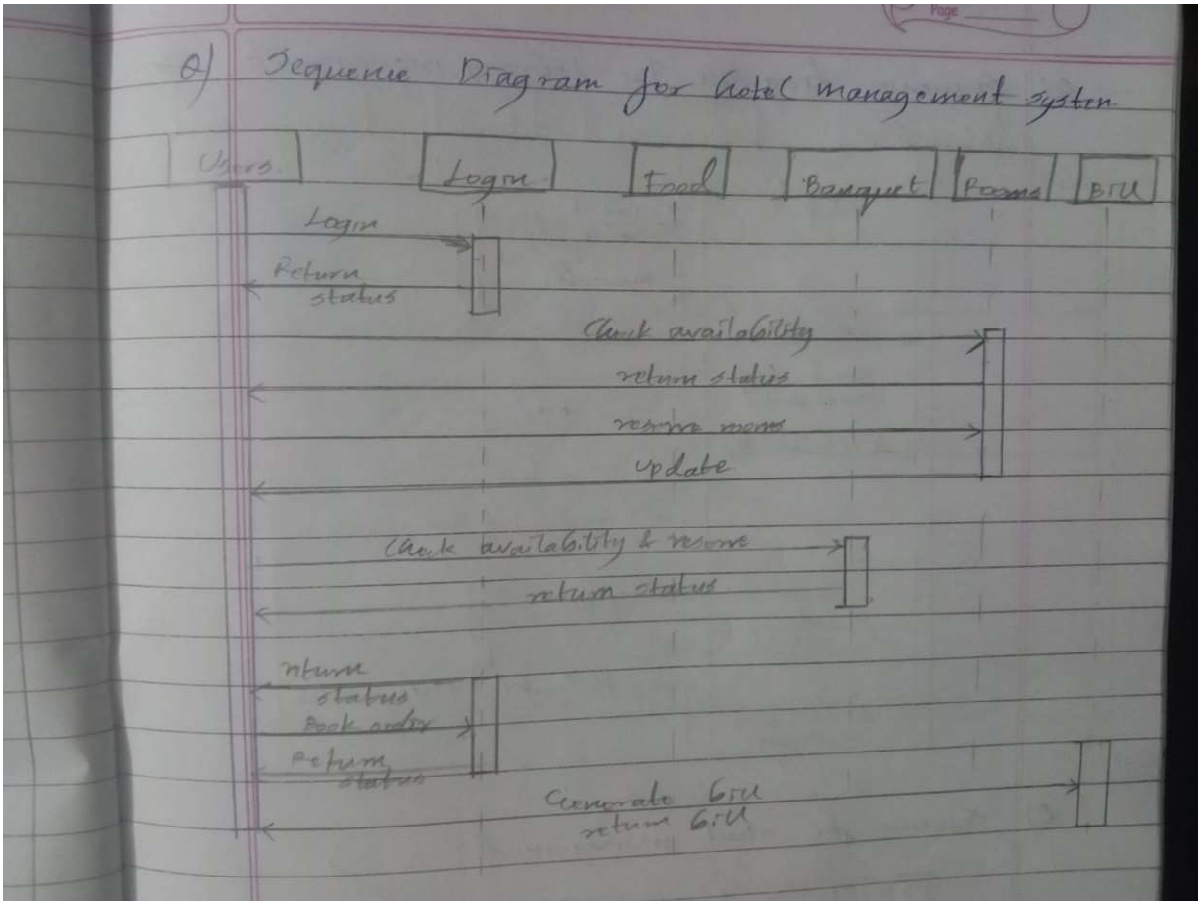


Fig 1.5



e. Activity Diagram:

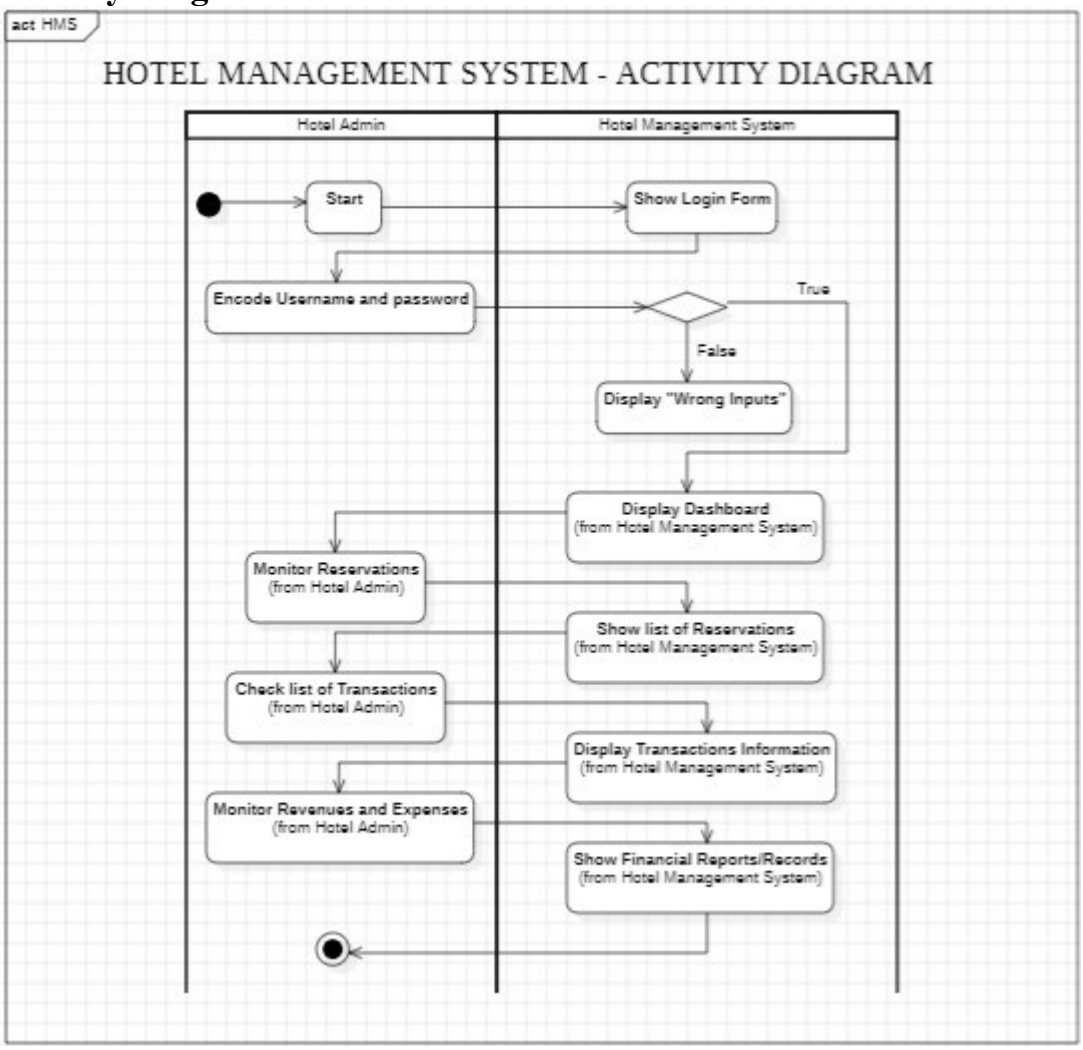
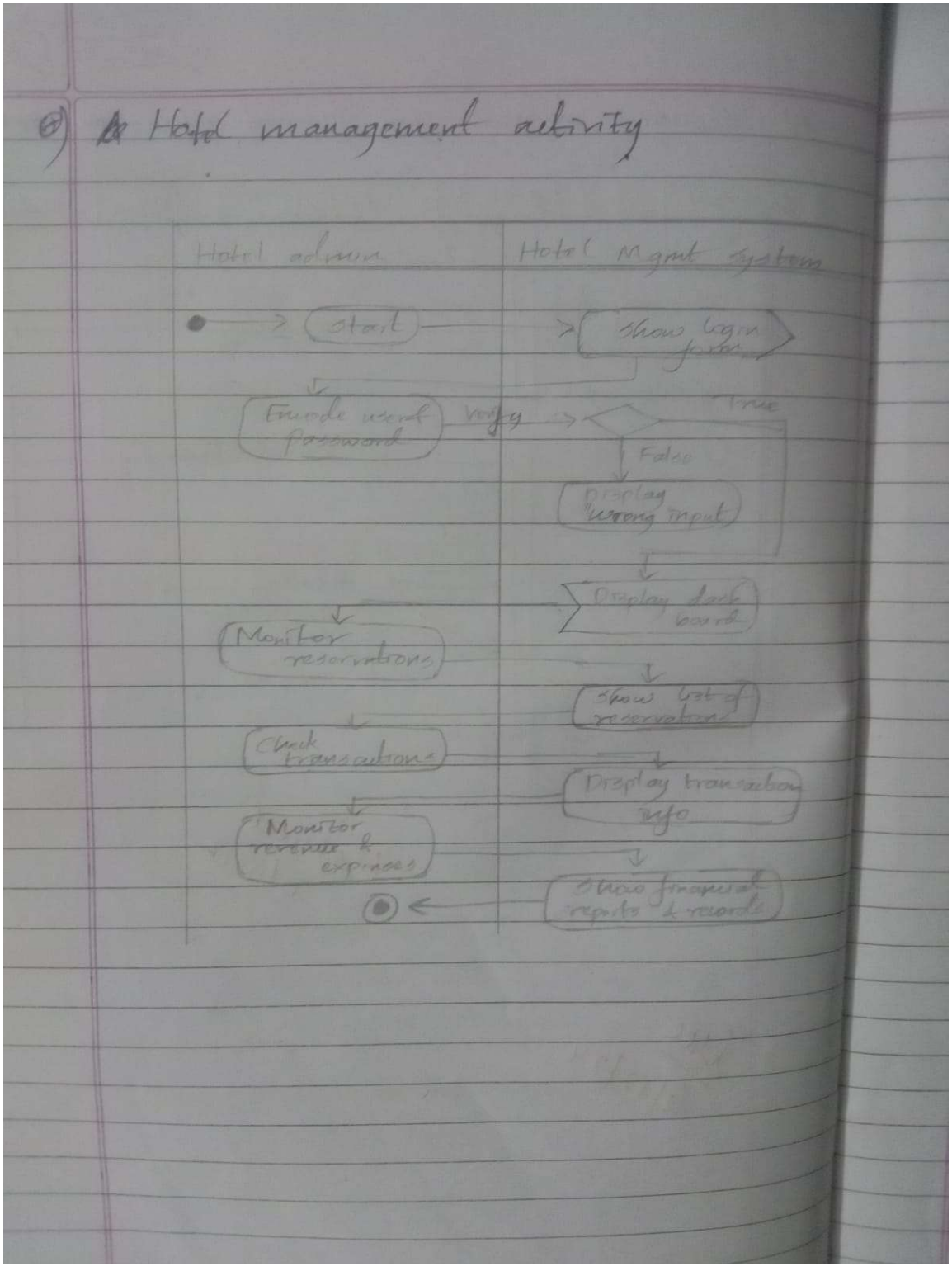
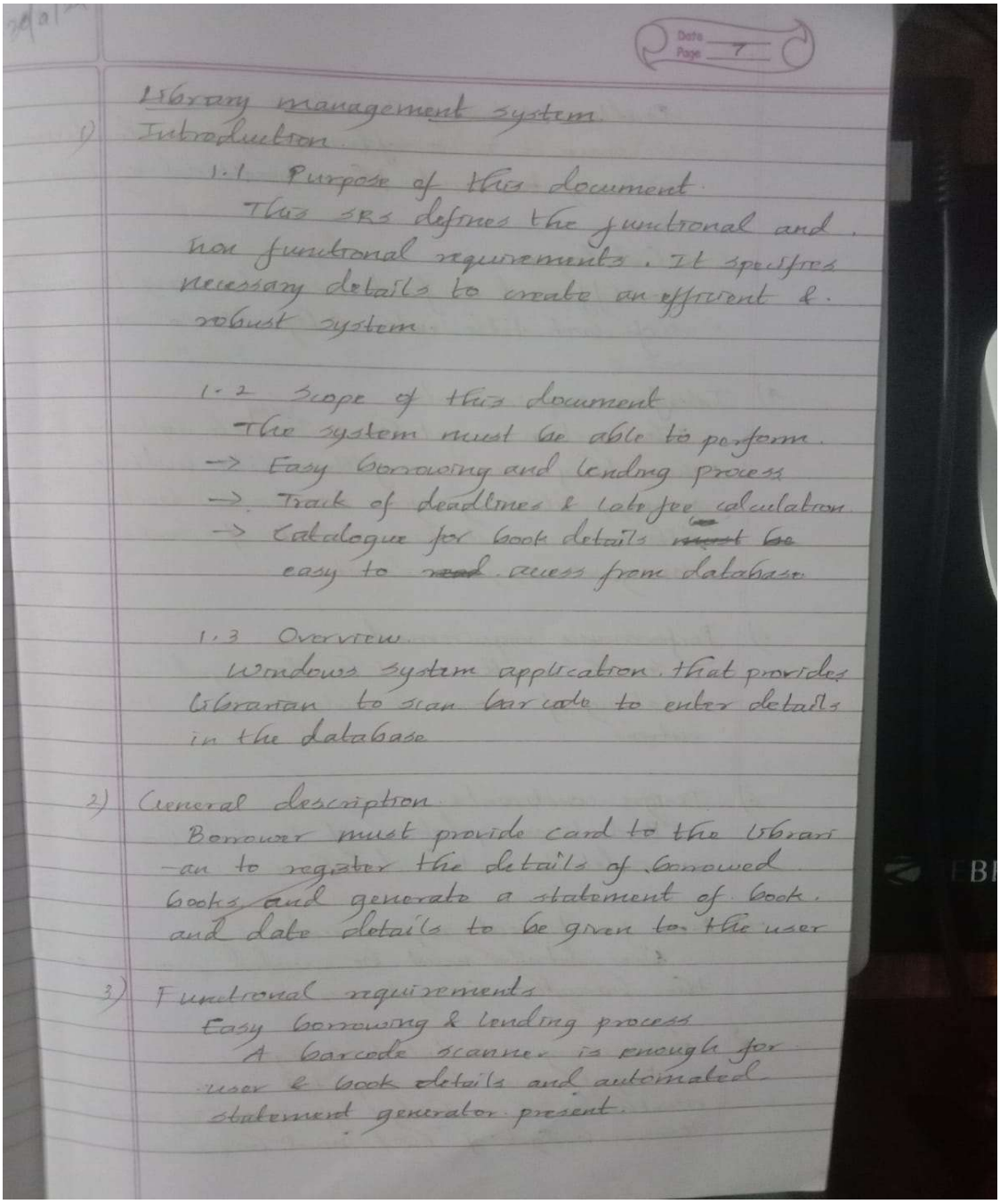


Fig 1.6


Hotel management activity
Hotel admin
Hotel Mgmt system
start
Show login
Enode user password
verify
True
False
Display wrong input
Display dash board
Monitor reservations
Show list of reservations
Check transactions
Display transaction info
Monitor rooms & expenses
Show financial reports & records

2. Library Management System

a. SRS Document:



Library management system.

1) Introduction

1.1 Purpose of this document

This SRS defines the functional and non functional requirements. It specifies necessary details to create an efficient & robust system.

1.2 Scope of this document

The system must be able to perform.

→ Easy borrowing and lending process

→ Track of deadlines & late fee calculation

→ Catalogue for book details ~~must~~ be easy to ~~read~~ access from database.

1.3 Overview.

Windows system application that provides Librarian to scan barcode to enter details in the database.

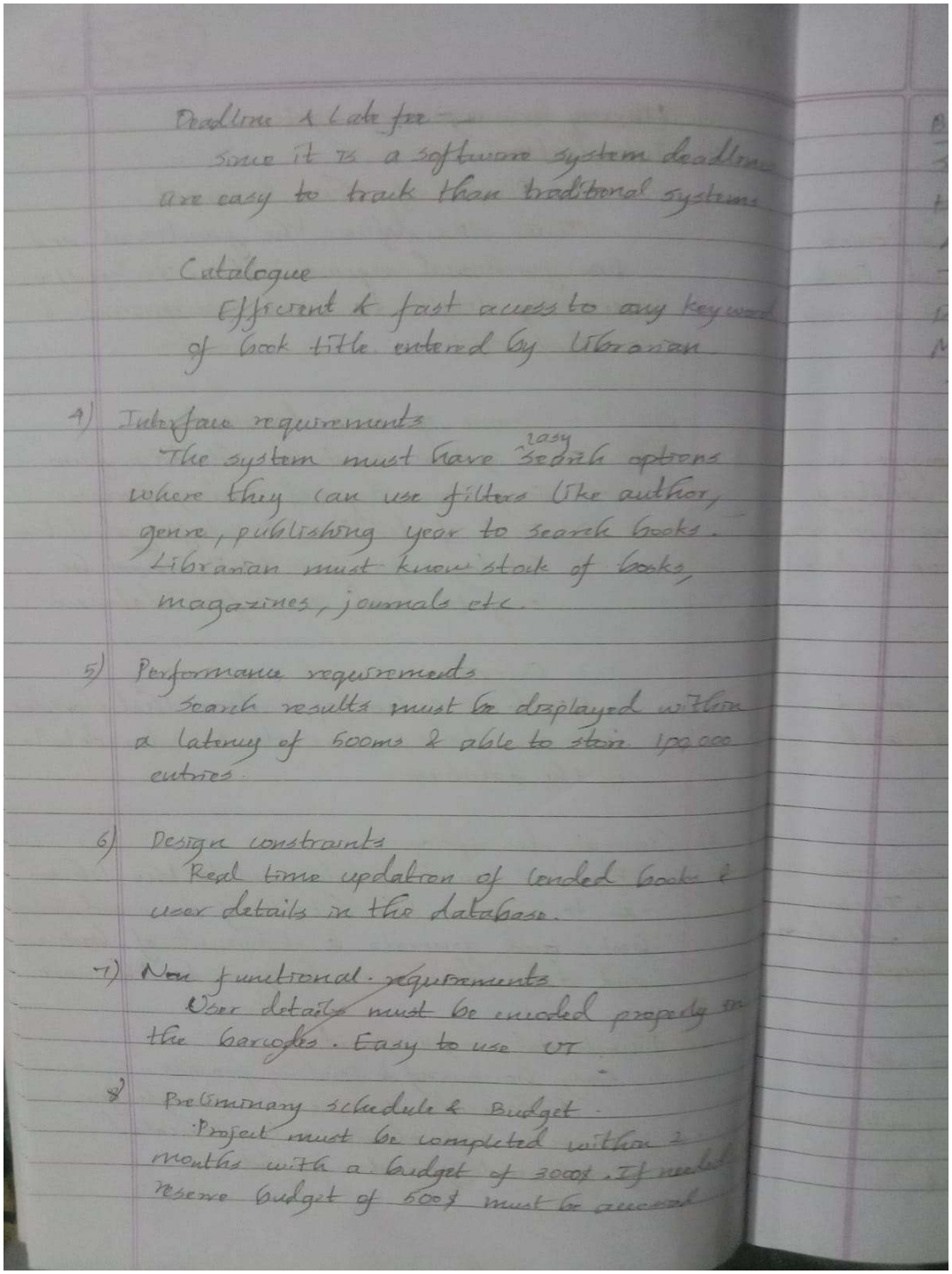
2) General description

Borrower must provide card to the Librarian to register the details of borrowed books and generate a statement of book and date details to be given to the user

3) Functional requirements

Easy borrowing & lending process

A barcode scanner is enough for user & book details and automated statement generator present.



Deadline & Late fee
Since it is a software system deadlines are easy to track than traditional systems

Catalogue
Efficient & fast access to any keyword of book title entered by librarian

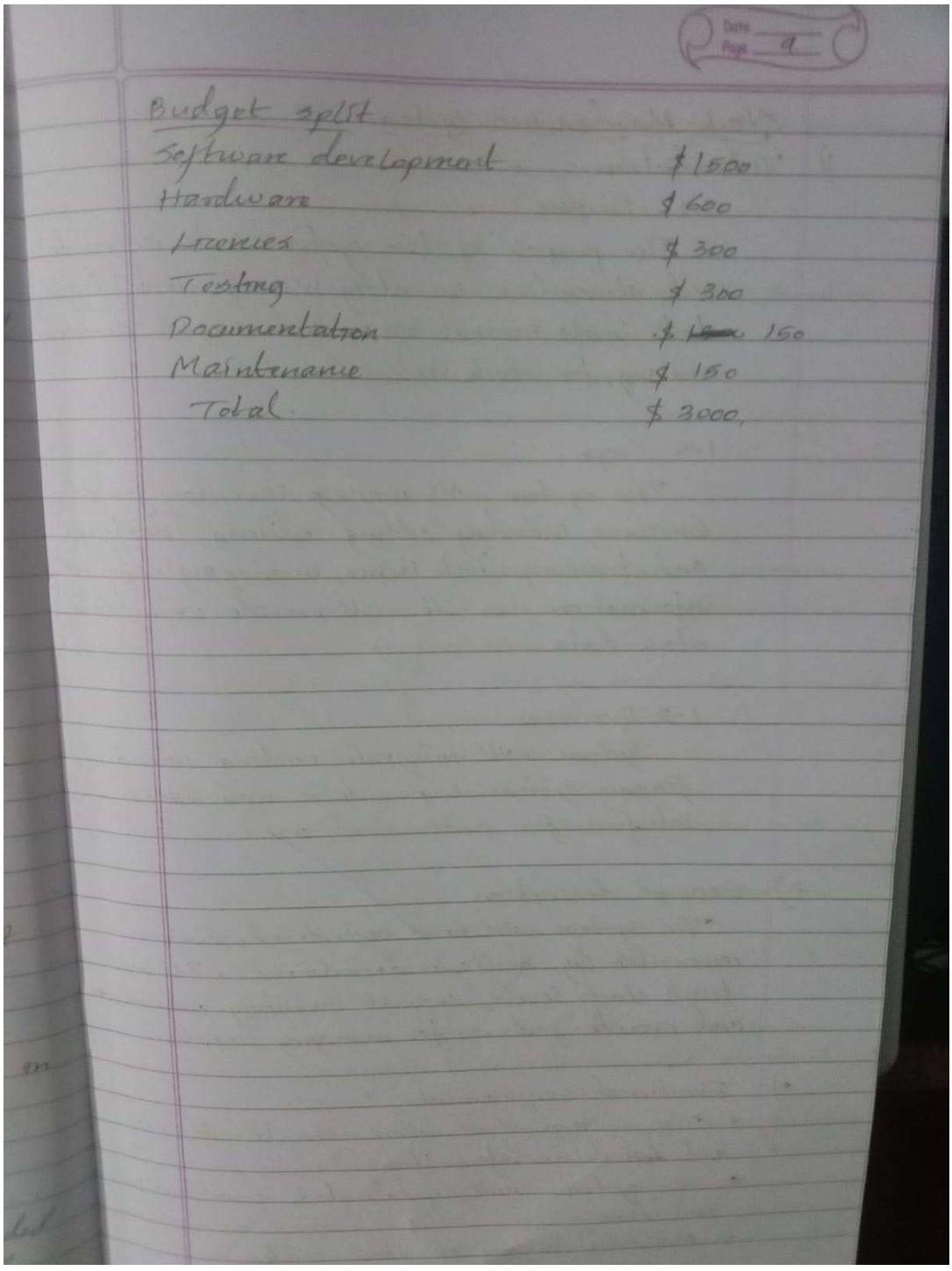
4) Interface requirements
The system must have easy search options where they can use filters like author, genre, publishing year to search books. Librarian must know stock of books, magazines, journals etc.

5) Performance requirements
Search results must be displayed within a latency of 500ms & able to store 100,000 entries.

6) Design constraints
Real time updation of lended book & user details in the database.

7) Non functional requirements
User details must be encoded properly on the barcodes. Easy to use UI

8) Preliminary schedule & Budget
Project must be completed within 2 months with a budget of 3000$. If needed reserve budget of 500$ must be allocated



Budget split

Software development	$1500
Hardware	$600
Licenses	$300
Testing	$300
Documentation	$150
Maintenance	$150
Total	$3000

b.Advanced Class Diagram:

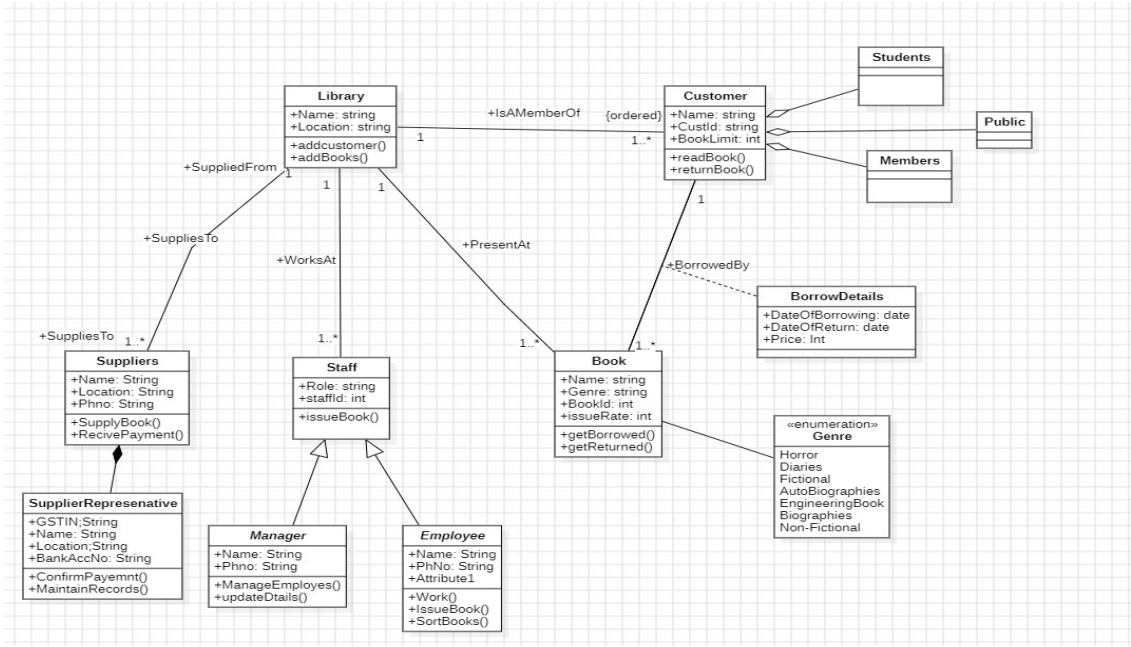
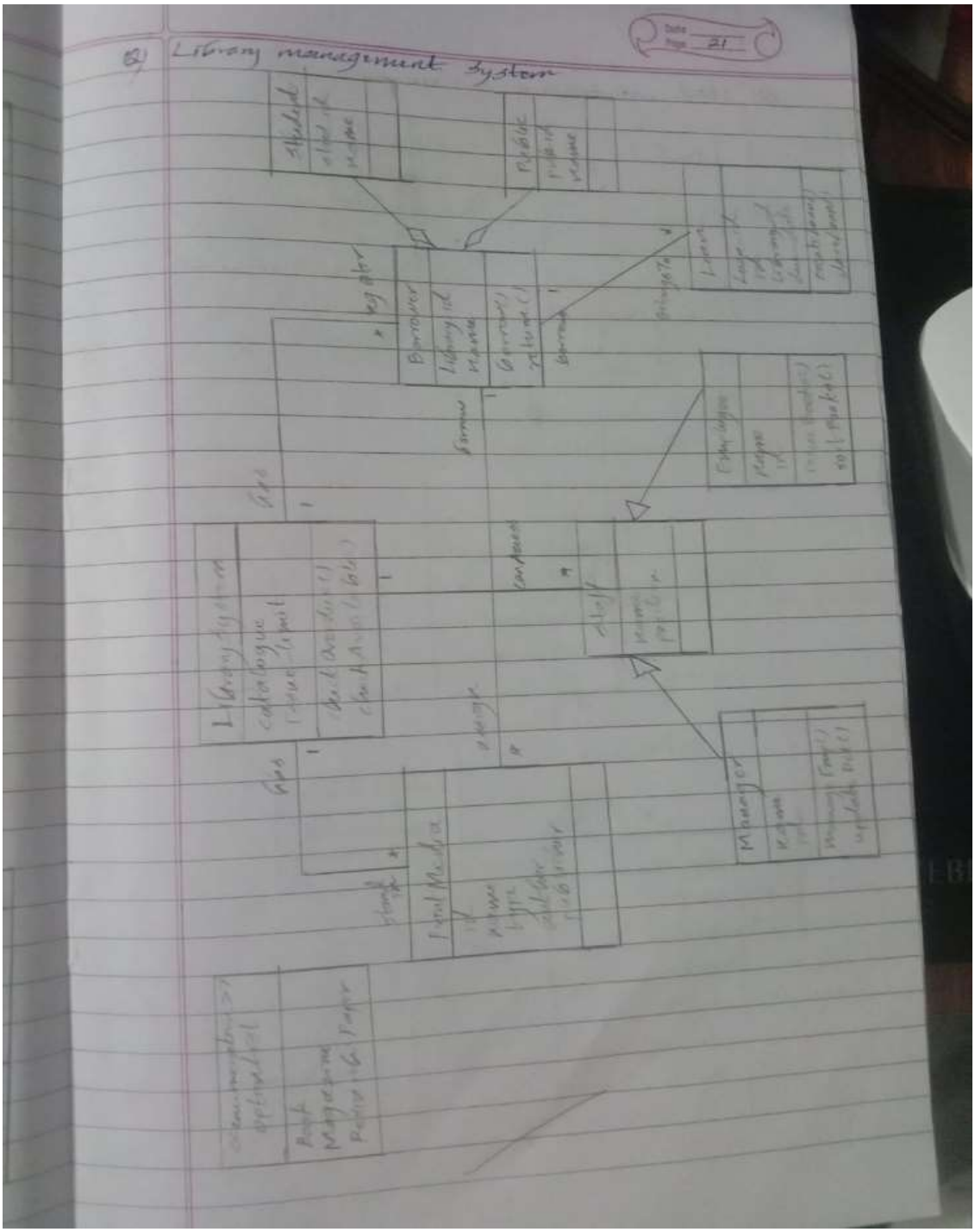


Fig 2.2:


Q) Library management system

c.Advanced State Diagram:

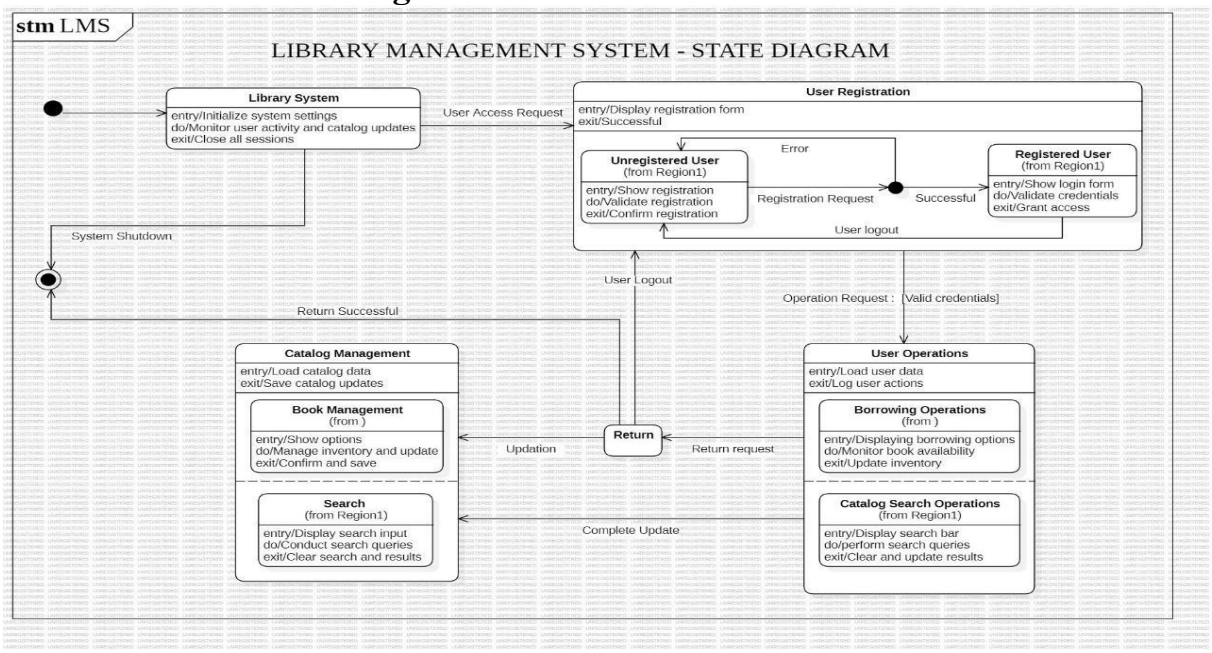
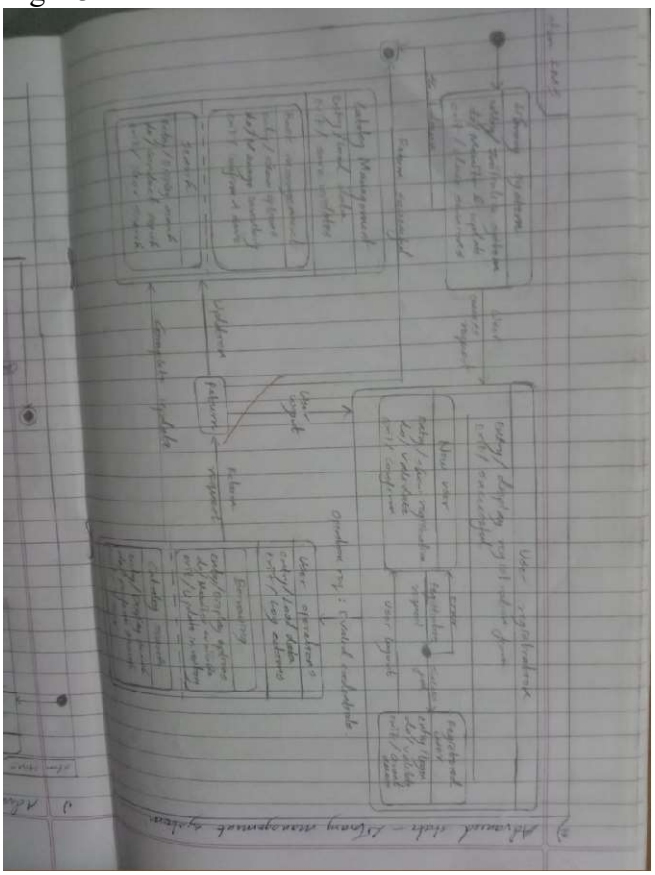


Fig 2.3:



d.Use Case Diagram:

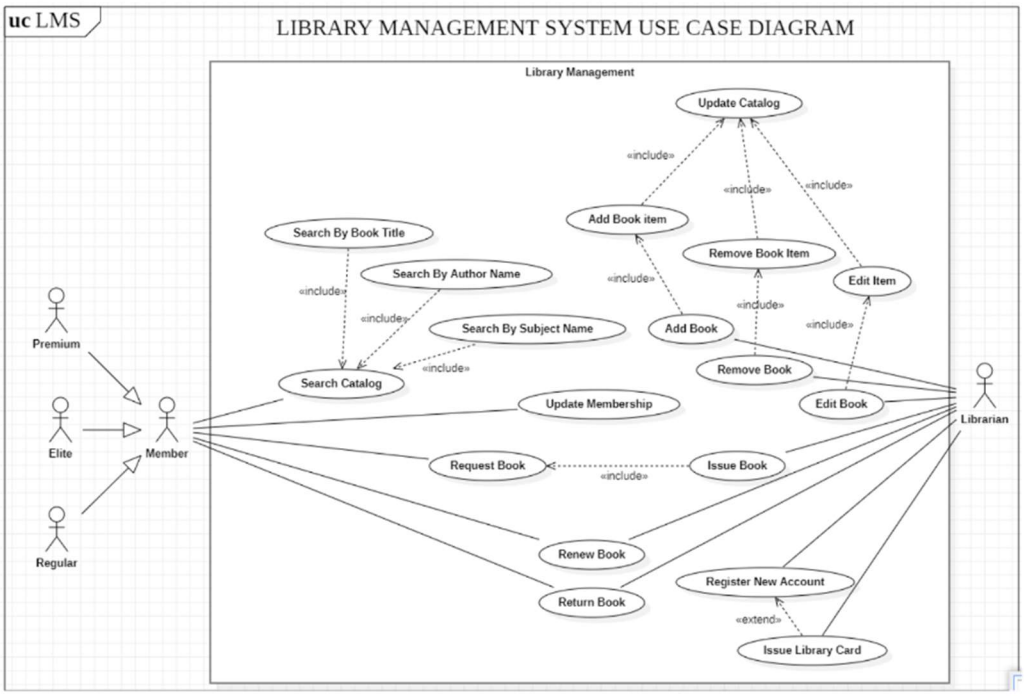
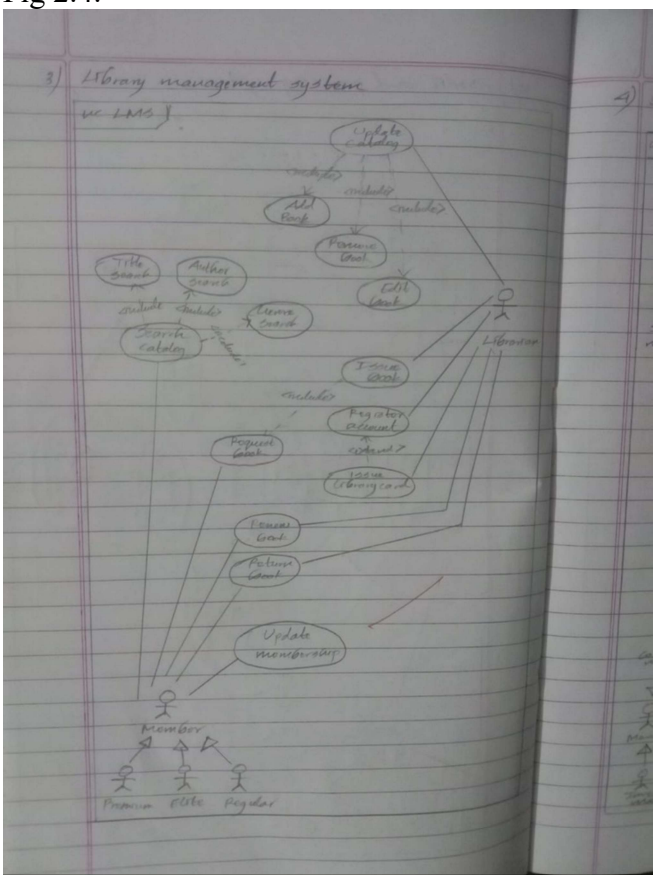
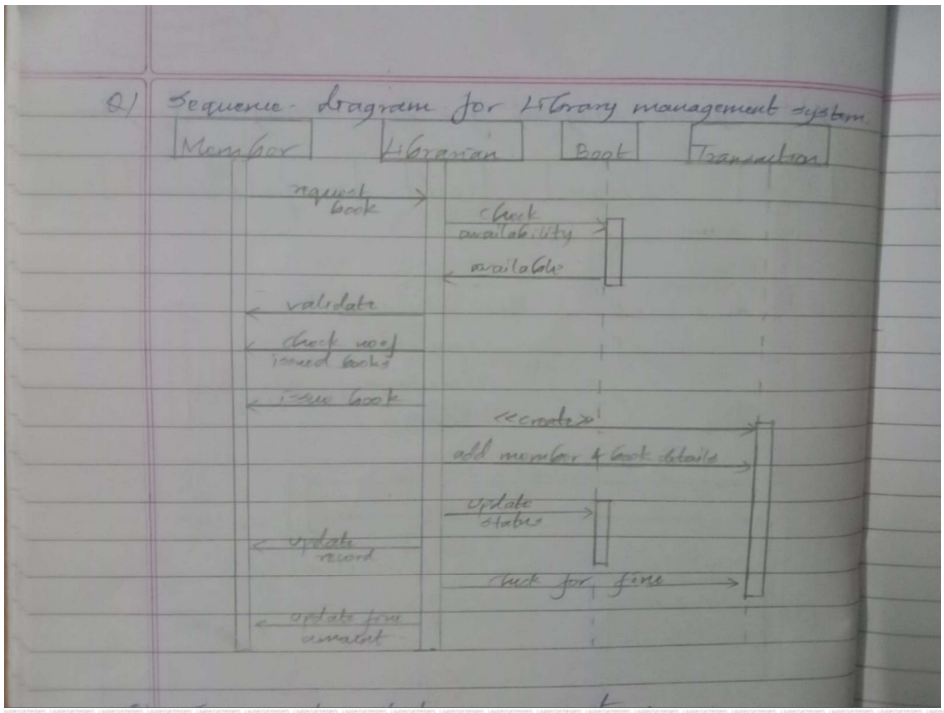


Fig 2.4:



e.Sequence Diagram:



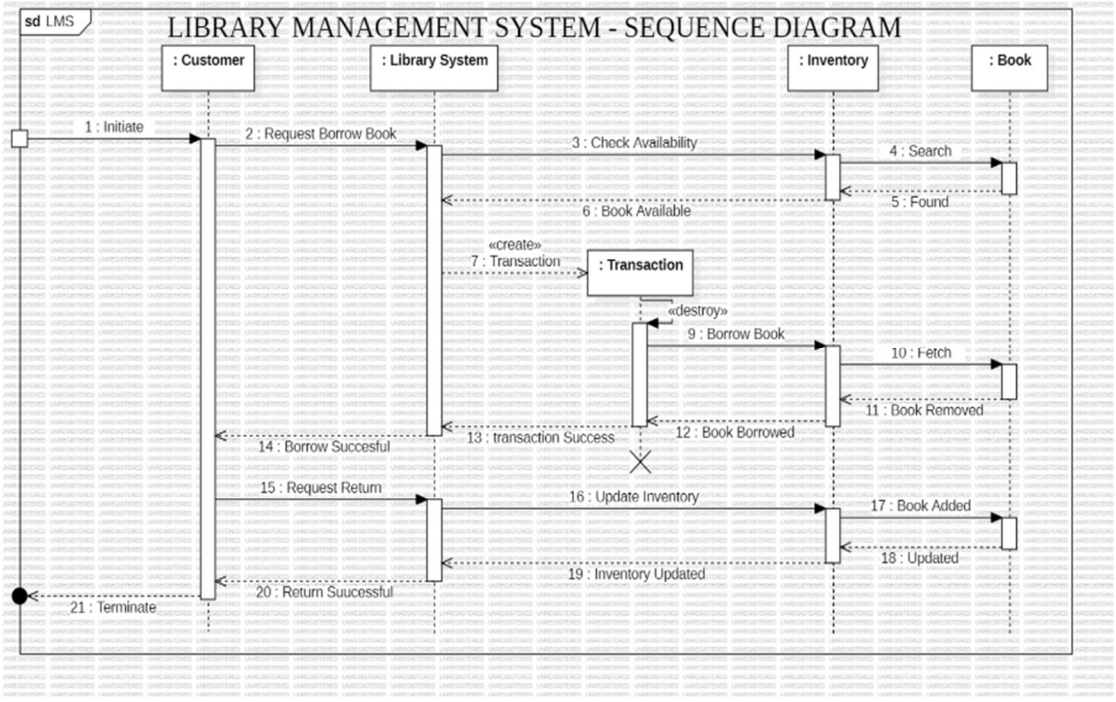
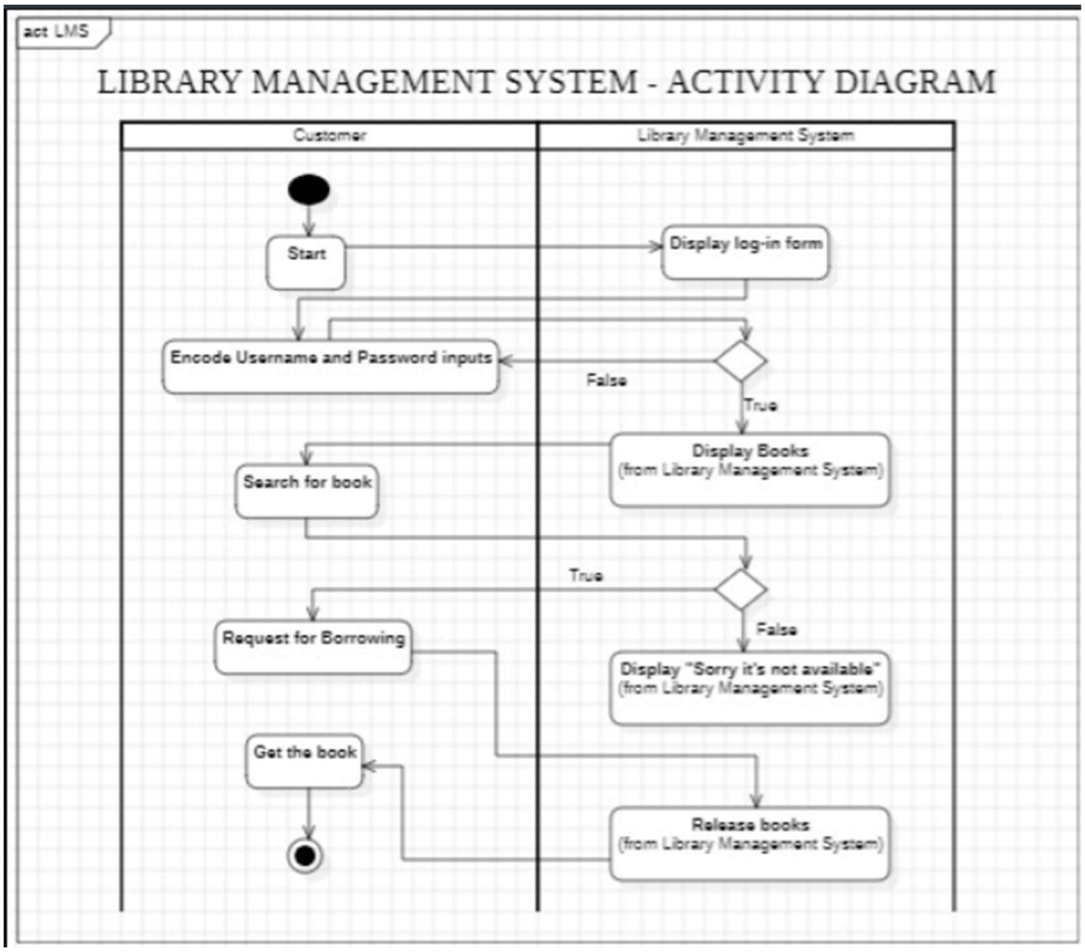


Fig 2.5:

f.Activity Diagram:



 Actors:

- **Customer**: Represents the user interacting with the system.
- **Library Management System**: Represents the system itself.

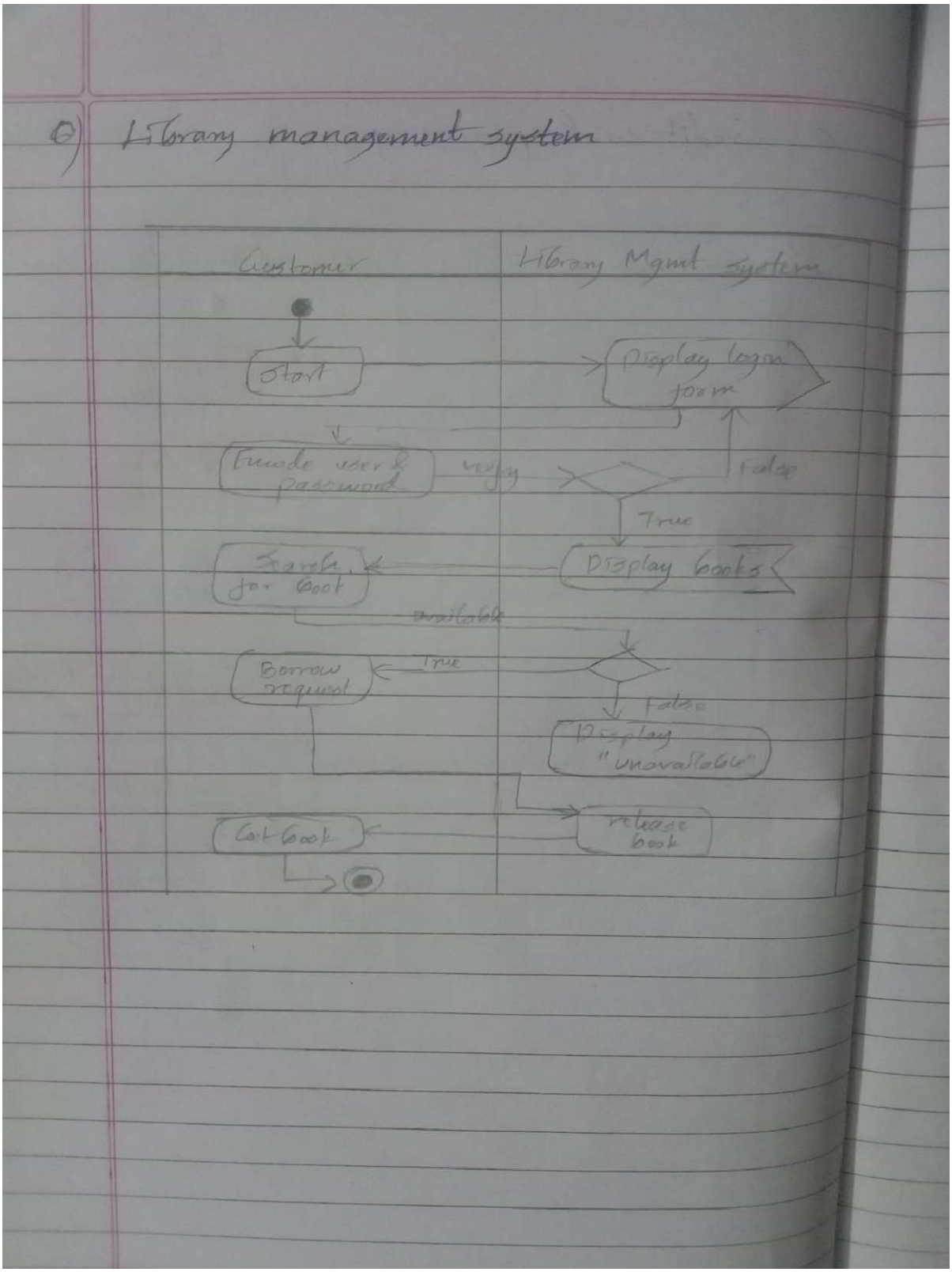
 Process Flow:

- The customer starts by logging in, encoding their username and password.
- The system validates the credentials. If invalid, the process stops; if valid,proceeds.
- The customer searches for a book. The system checks the book's availability:
    - If the book is unavailable, the system displays a "not available" message.
    - If available, the customer requests to borrow the book, and the system releases it to them.

 Decisions:

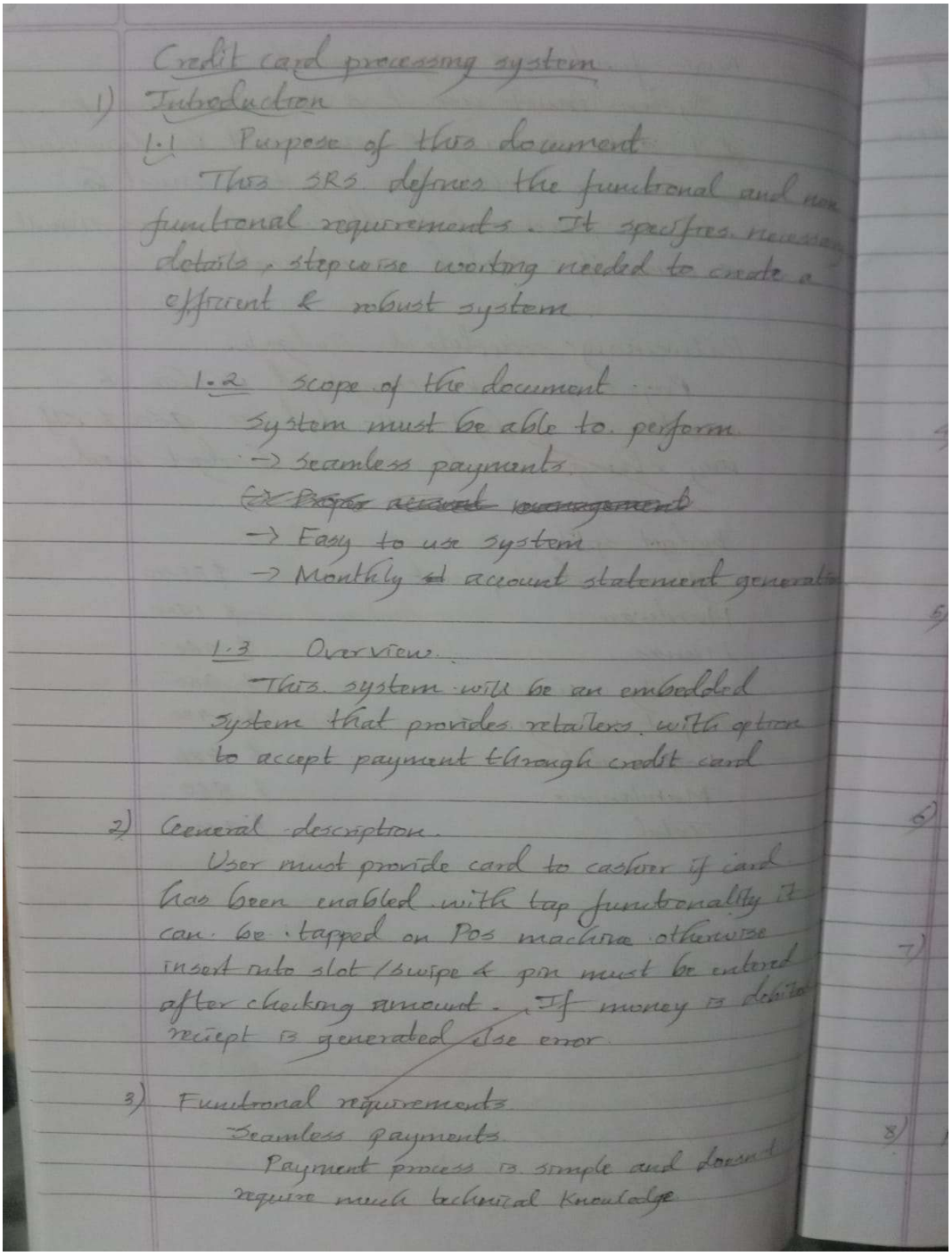
- There are decision points for login validation and book availability, indicating alternate flows.

6) Library management system



3. Credit Card Management System

a. SRS Document:



Credit card processing system

1) Introduction

1.1 Purpose of this document

This SRS defines the functional and non functional requirements. It specifies necessary details, stepwise working needed to create a efficient & robust system

1.2 Scope of the document

System must be able to perform

-> Seamless payments

~~-> Proper record management~~

-> Easy to use system

-> Monthly ~~d~~ account statement generate

1.3 Overview

This system will be an embedded system that provides retailers with option to accept payment through credit card

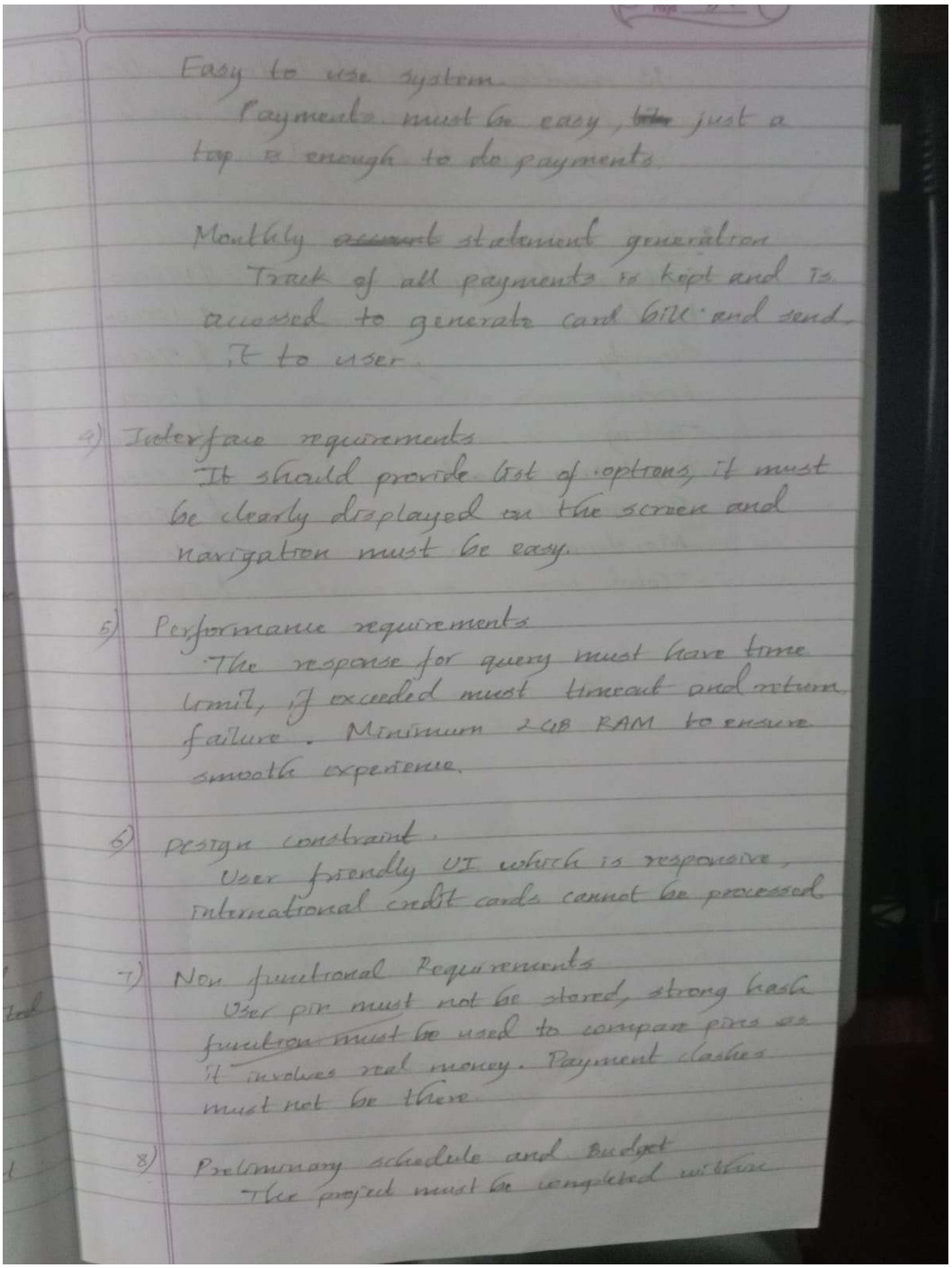
2) General description

User must provide card to cashier if card has been enabled with tap functionality it can be tapped on Pos machine otherwise insert into slot / swipe & pin must be entered after checking amount. If money is debited receipt is generated else error

3) Functional requirements

Seamless payments

Payment process is simple and doesn't require much technical knowledge



Easy to use system.
Payments must be easy, just a tap is enough to do payments

Monthly statement generation
Track of all payments is kept and is accessed to generate card bill and send it to user.

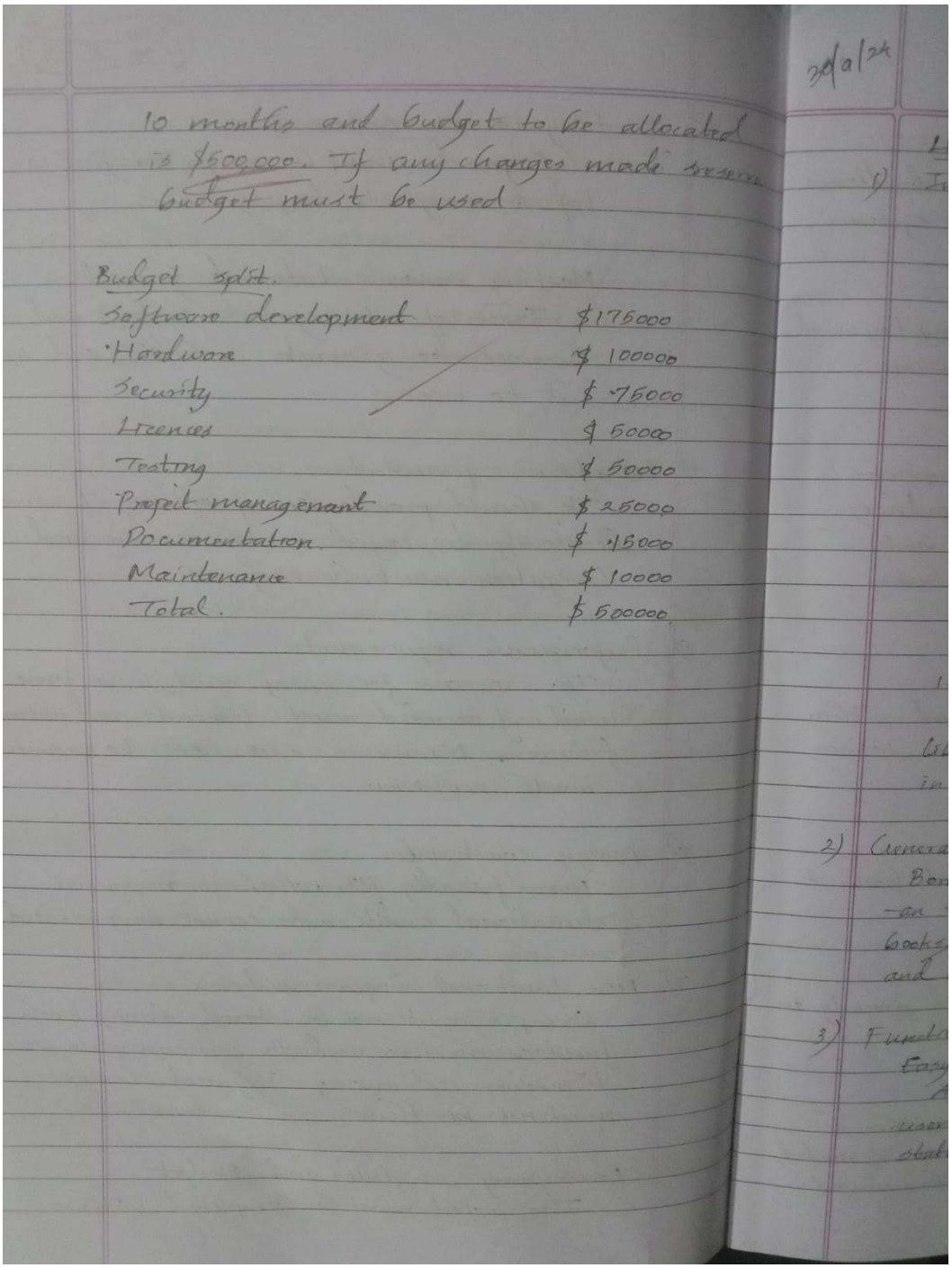
4) Interface requirements
It should provide list of options, it must be clearly displayed on the screen and navigation must be easy.

5) Performance requirements
The response for query must have time limit, if exceeded must timeout and return failure. Minimum 2GB RAM to ensure smooth experience.

6) Design constraint.
User friendly UI which is responsive, International credit cards cannot be processed.

7) Non functional Requirements
User pin must not be stored, strong hash function must be used to compare pins as it involves real money. Payment clashes must not be there.

8) Preliminary schedule and Budget
The project must be completed within



10 months and budget to be allocated is $500,000. If any changes made budget must be used

Budget split.

Software development	$175000
Hardware	$ 100000
Security	$ 75000
Licences	$ 50000
Testing	$ 50000
Projet management	$ 25000
Documentation	$ 15000
Maintenance	$ 10000
Total.	$ 500000

b. Advanced Class Diagram:

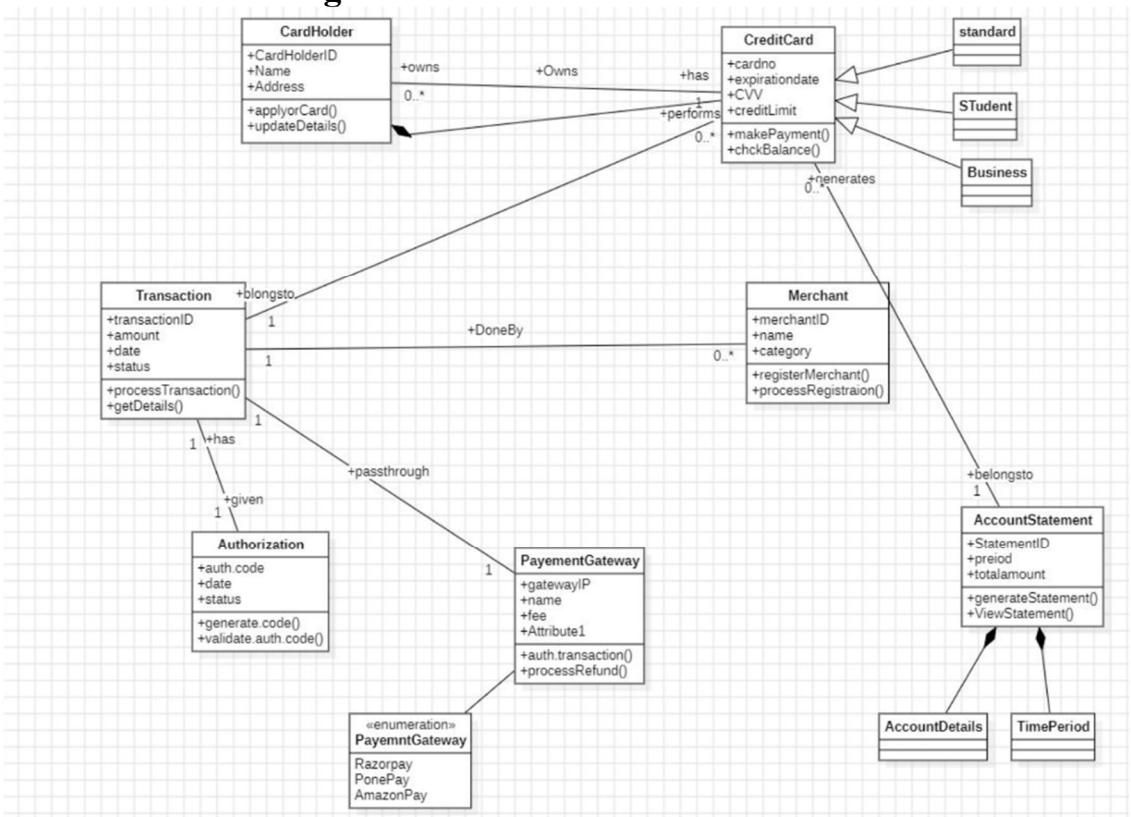
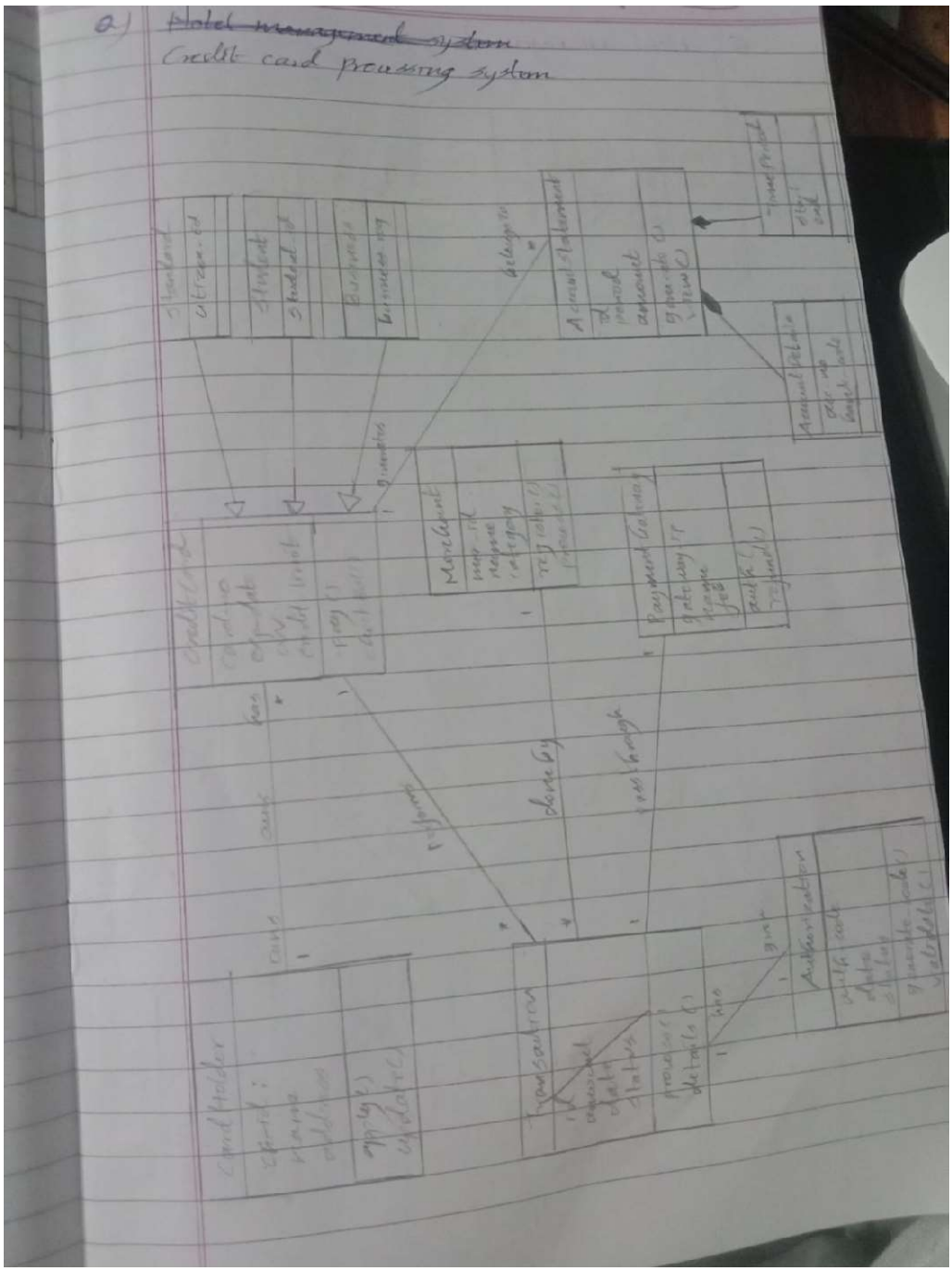


Fig 3.1:

The main entities include:

- Card Holder, who owns a Credit Card used for transactions. The credit card has details like card number, CVV, and credit limit, with operations for payments and balance checks.
- Transaction, which records details like amount, date, and status, is linked to both Authorization (for validation) and Payment Gateway (for processing or refunding payments).
- Merchant, which registers businesses accepting payments and processes transactions from customers.
- Account Statement, which tracks transaction history, generates statements, and is linked to periods and account details.

The diagram also shows inheritance, with specialized credit cards (e.g., Student or Business), and enumerates multiple payment gateways (e.g., RazorPay, PhonePay). It maps how users, cards, merchants, and gateways interact in the system to process payments seamlessly.


2) Hotel management system
Credit card processing system

c. Advanced State Diagram:

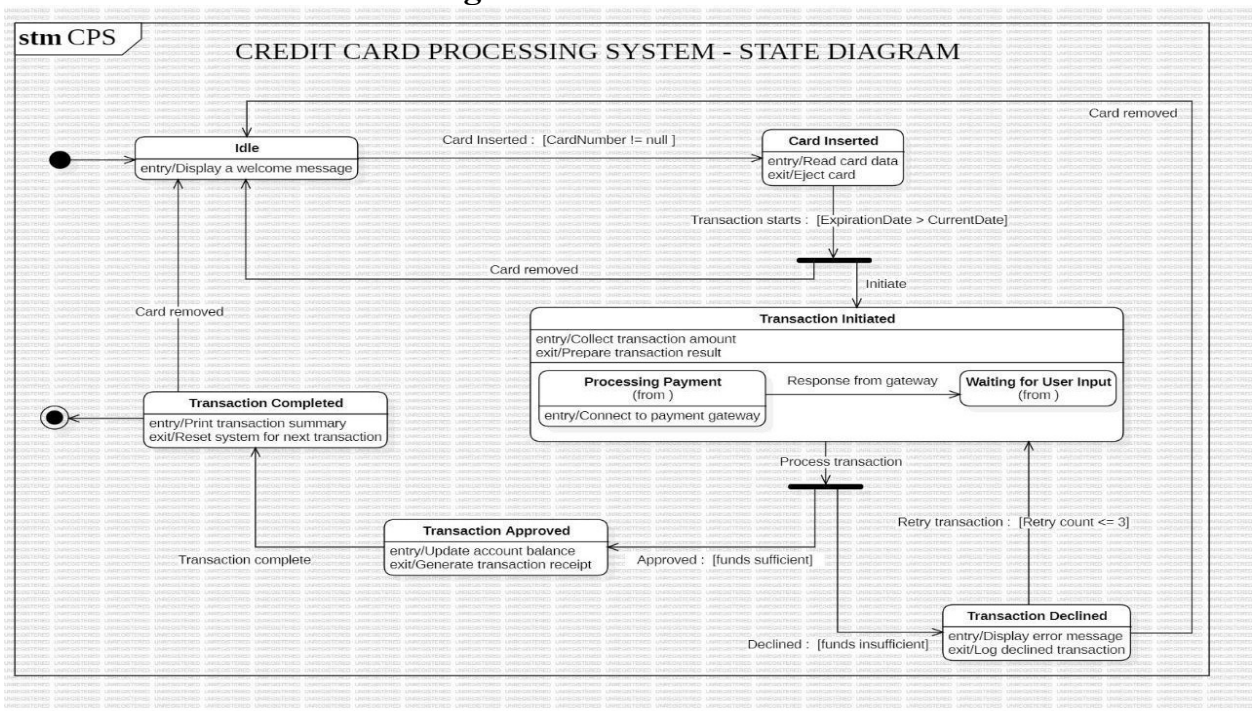
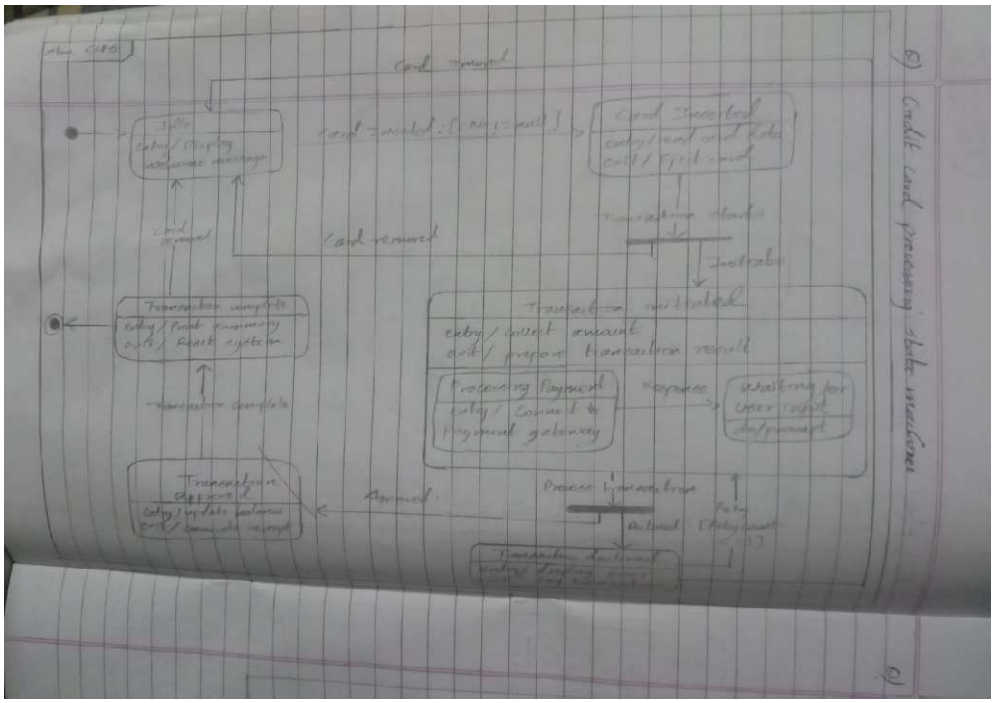


Fig 3.2:



d. Use Case Diagram:

Credit Processing System

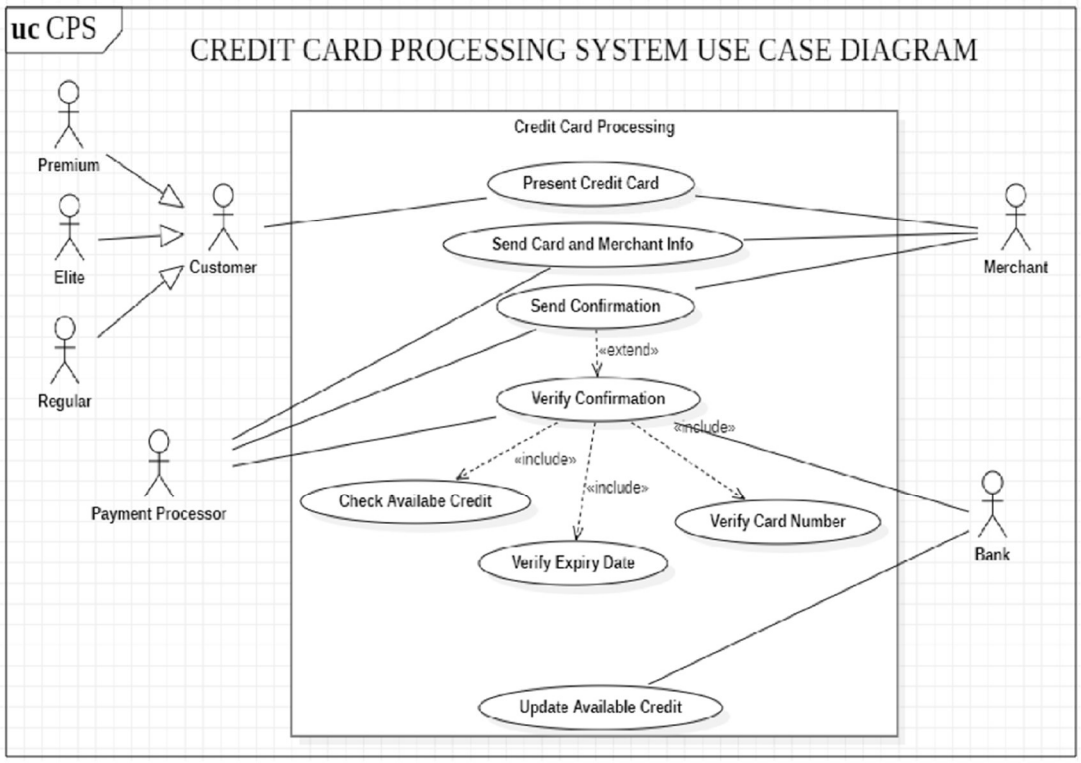
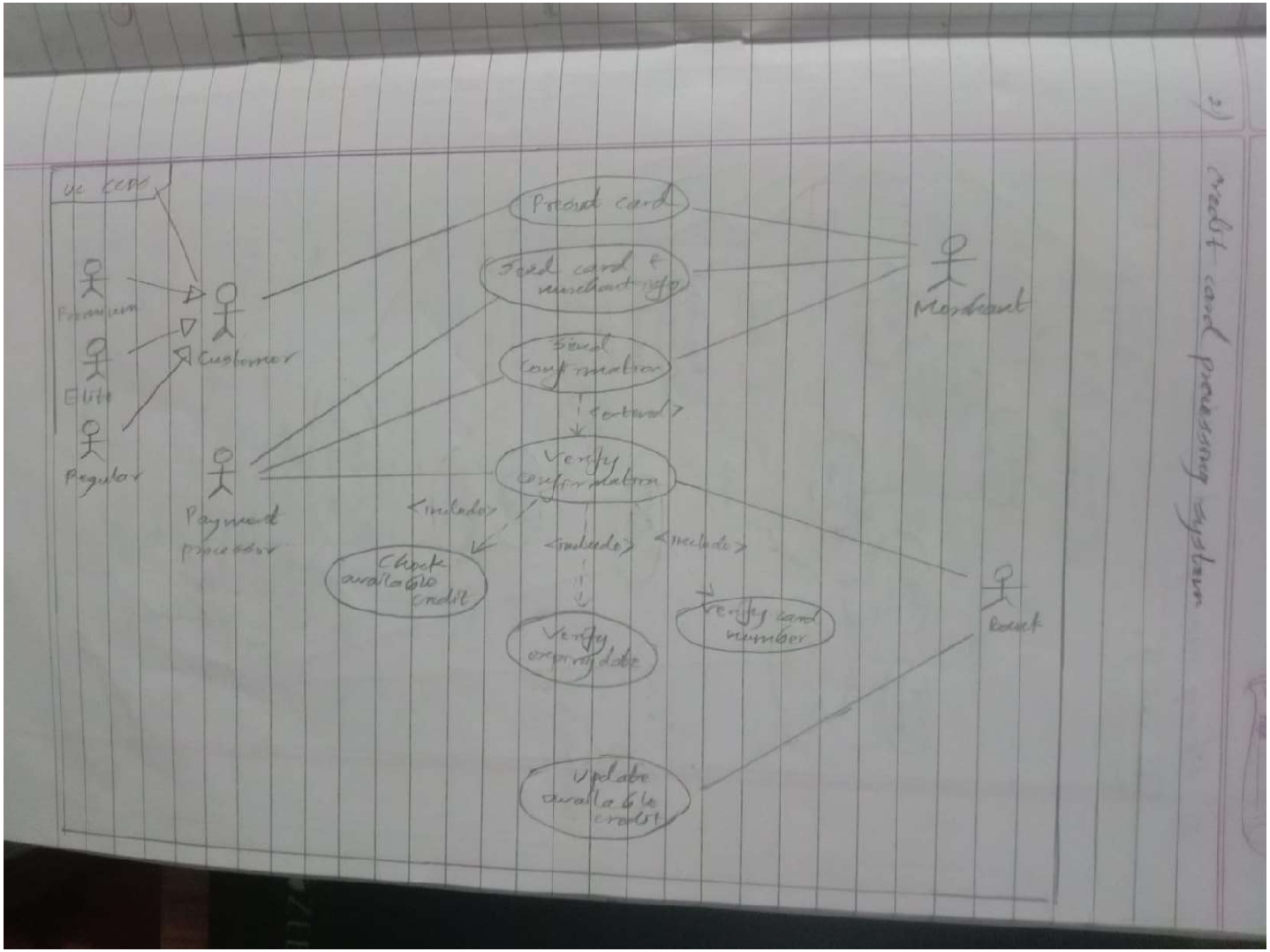


Fig 3.3:


Use cases
Premium
Elite
Regular
Customer
Payment processor
Provide card
Feed card & merchant info
Send confirmation
Verify information
Check available credit
Verify expiry date
Verify card number
Update available credit
Merchant
Bank
<extend>
<include>
<include>
<include>
credit card processing system

e. Sequence Diagram:

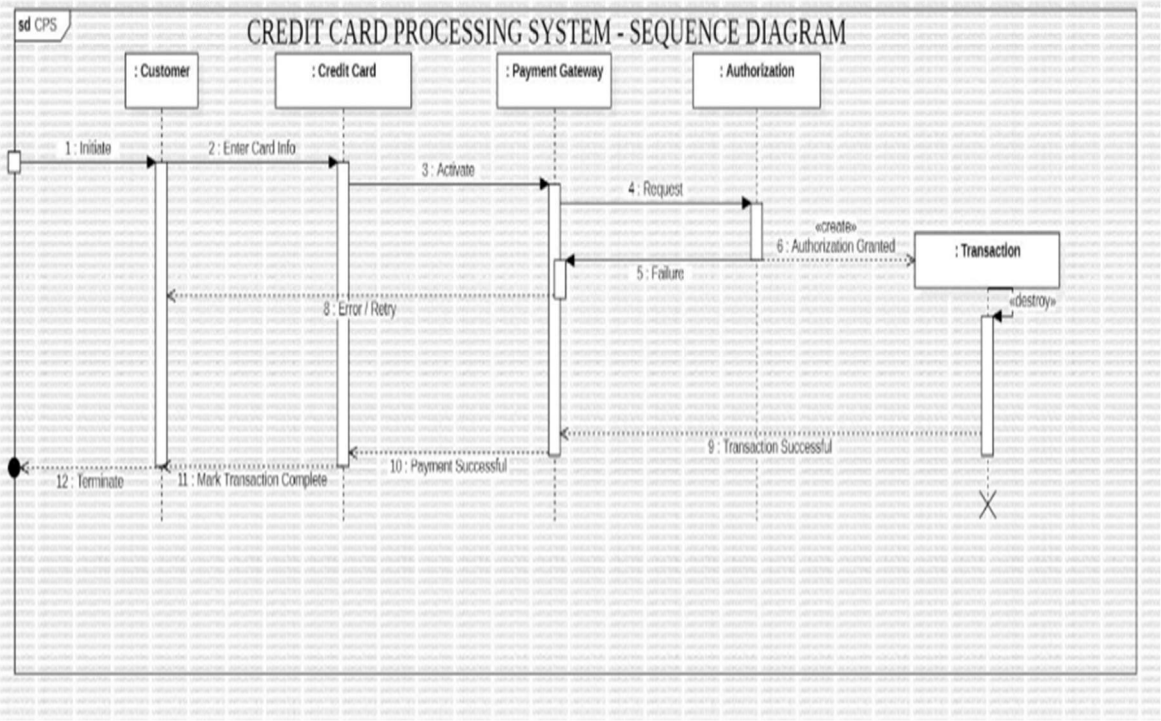
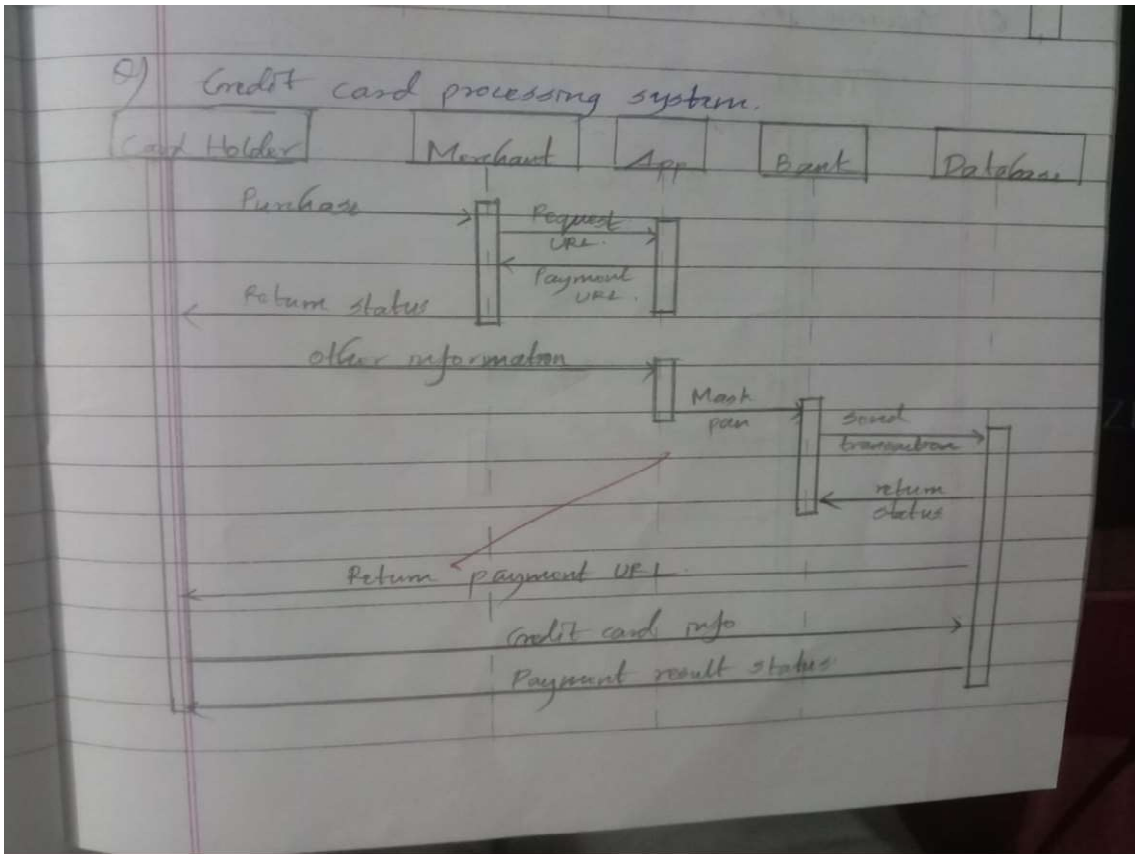


Fig 3.4:



f. Activity Diagram:

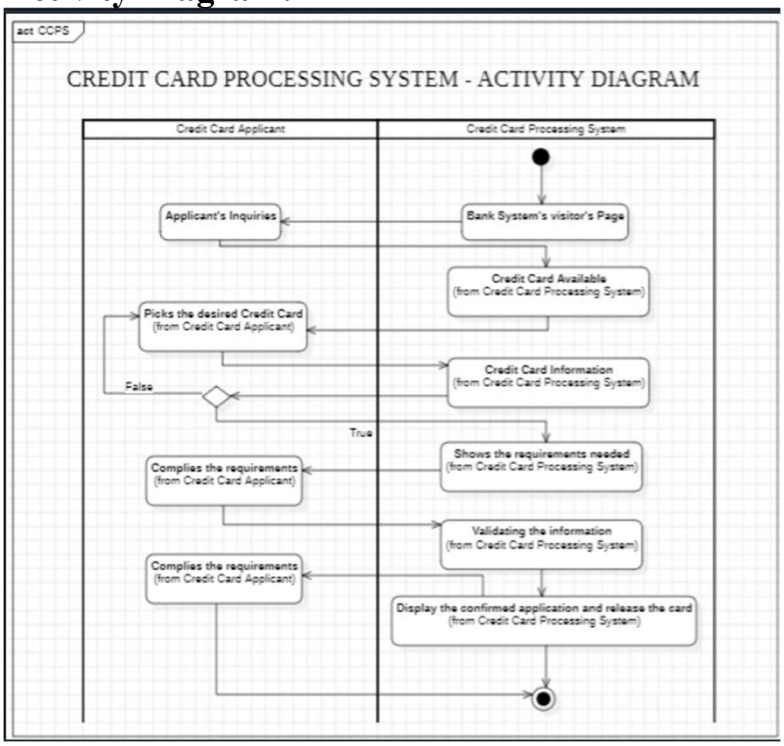


Fig 3.5:

Applicant's Journey:

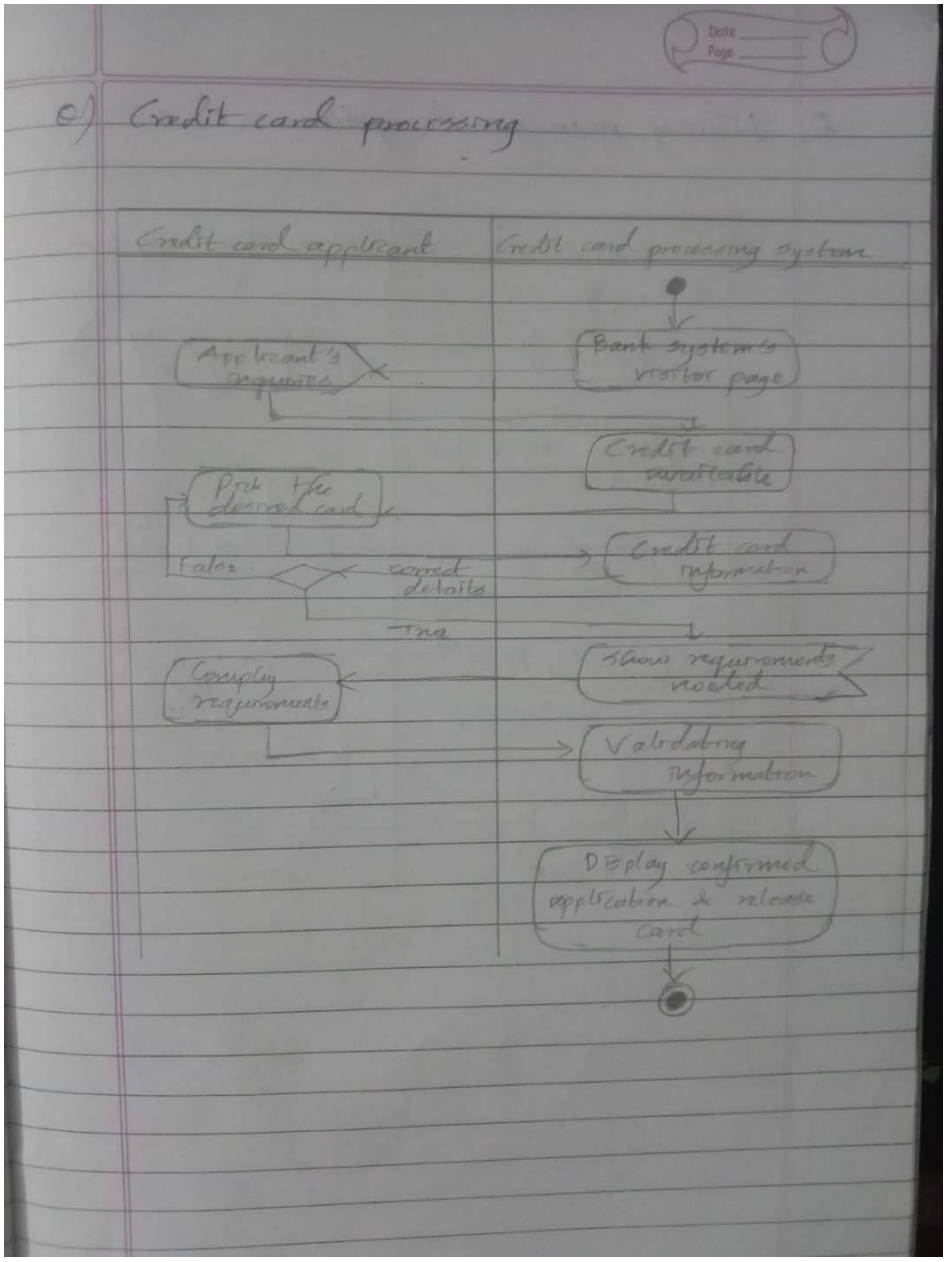
- The process starts with the applicant making inquiries and selecting a desired credit card.
- If the chosen card is available, the applicant compiles and submits the required documents.

Processing System's Role:

- The system displays the bank's visitor page and shows the available cards and their details.
- It provides the list of requirements needed and validates the submitted information.
- Upon successful validation, the system confirms the application and releases the card.

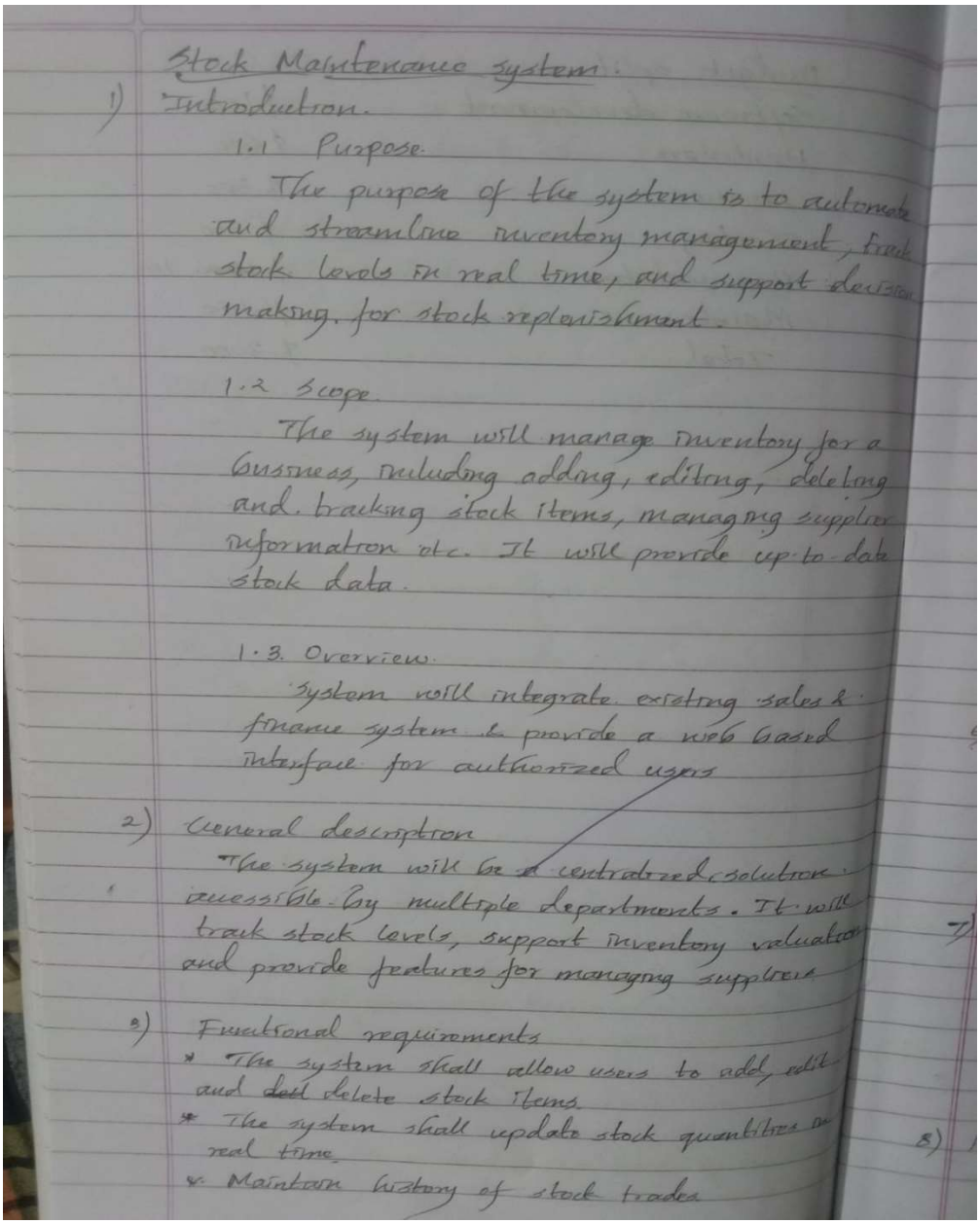
Decision Point:

- If the card is unavailable, the applicant must re-select or terminate the process.


Date
Page
e) Credit card processing
Credit card applicant
Credit card processing system
Applicant's requires
Bank systems visitor page
Credit card available
Pick the desired card
Credit card information
False
correct details
True
Show requirements needed
Comply requirements
Validating information
Display confirmed application & release card

4. Stock Maintenance system

a. SRS Document:



Stock Maintenance system:

1) Introduction.

1.1 Purpose.

The purpose of the system is to automate and streamline inventory management, track stock levels in real time, and support decision making for stock replenishment.

1.2 Scope.

The system will manage inventory for a business, including adding, editing, deleting and tracking stock items, managing supplier information etc. It will provide up-to-date stock data.

1.3 Overview.

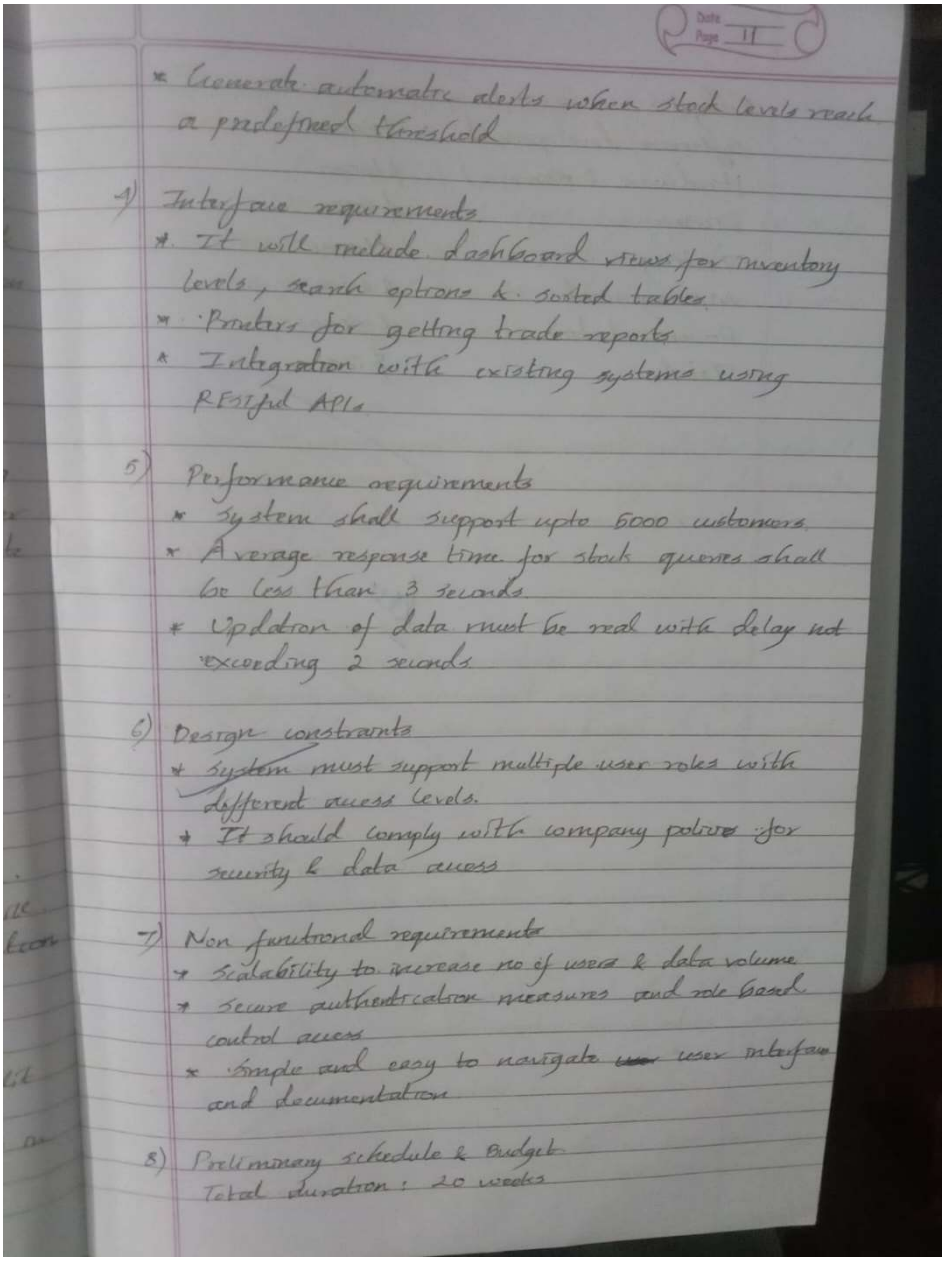
System will integrate existing sales & finance system & provide a web based interface for authorized users

2) General description

The system will be a centralized solution accessible by multiple departments. It will track stock levels, support inventory valuation and provide features for managing suppliers

3) Functional requirements

* The system shall allow users to add, edit and ~~del~~ delete stock items
* The system shall update stock quantities in real time
* Maintain history of stock trades


* Generate automatic alerts when stock levels reach a predefined threshold

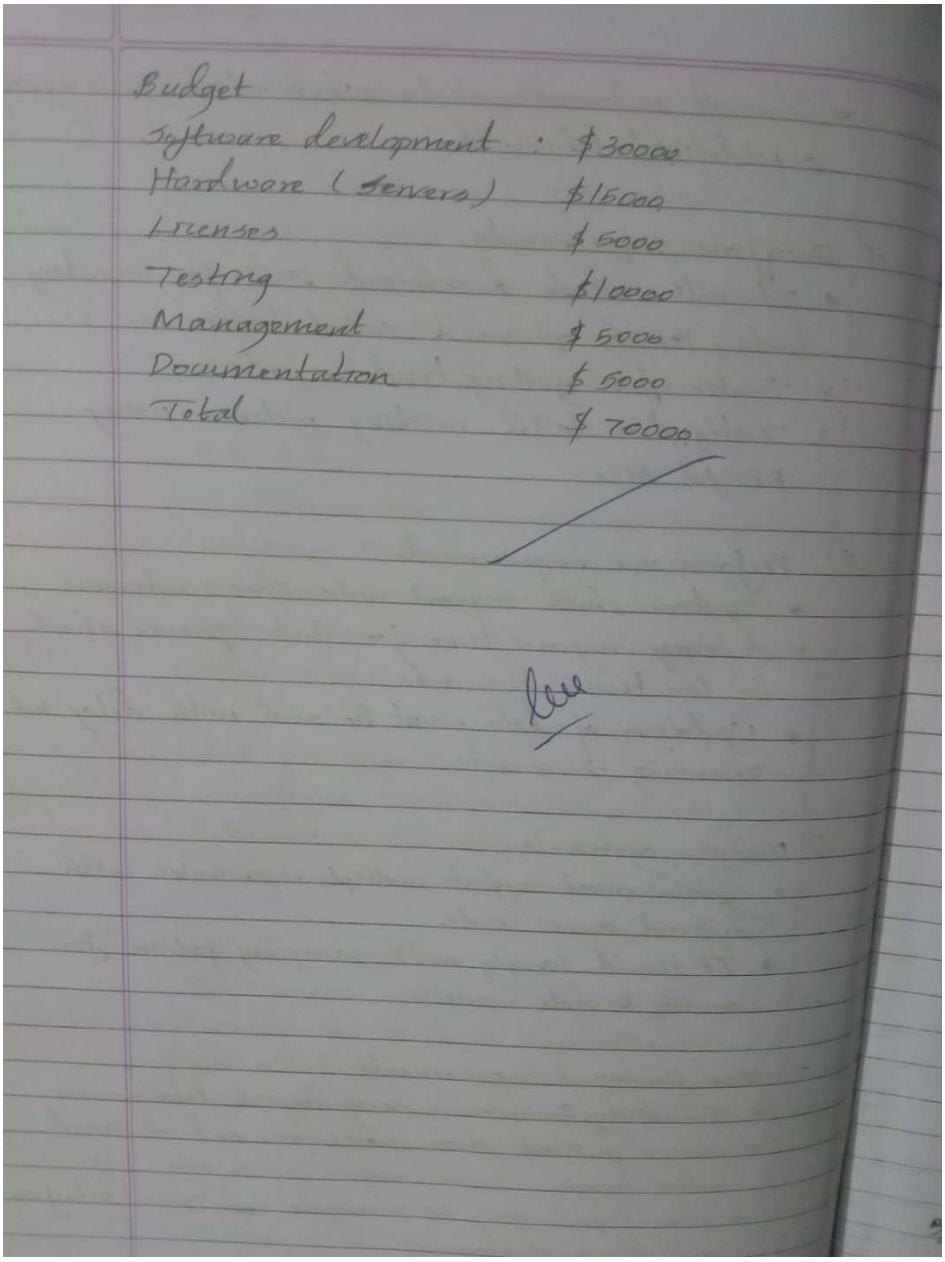
4) Interface requirements
* It will include dashboard views for inventory levels, search options & sorted tables.
* Printers for getting trade reports
* Integration with existing systems using RESTful APIs

5) Performance requirements
* System shall support upto 6000 customers.
* Average response time for stock queries shall be less than 3 seconds.
* Updation of data must be real with delay not exceeding 2 seconds

6) Design constraints
* System must support multiple user roles with different access levels.
* It should comply with company polices for security & data access

7) Non functional requirements
* Scalability to increase no of users & data volume
* Secure authentication measures and role based control access
* Simple and easy to navigate user interface and documentation

8) Preliminary schedule & Budget
Total duration: 20 weeks



Budget

Software development :	$30000
Hardware (Servers)	$15000
Licenses	$5000
Testing	$10000
Management	$5000
Documentation	$5000
Total	$70000

b.Advanced Class Diagram:

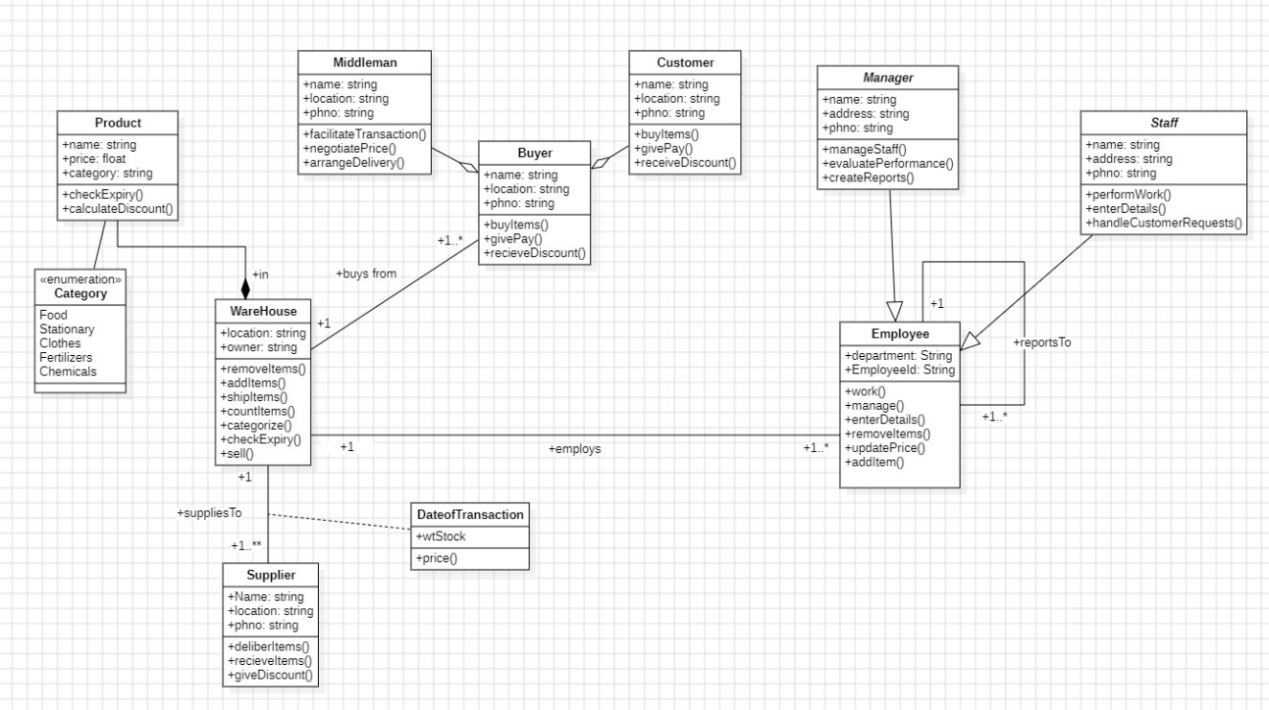


Fig 4.1:

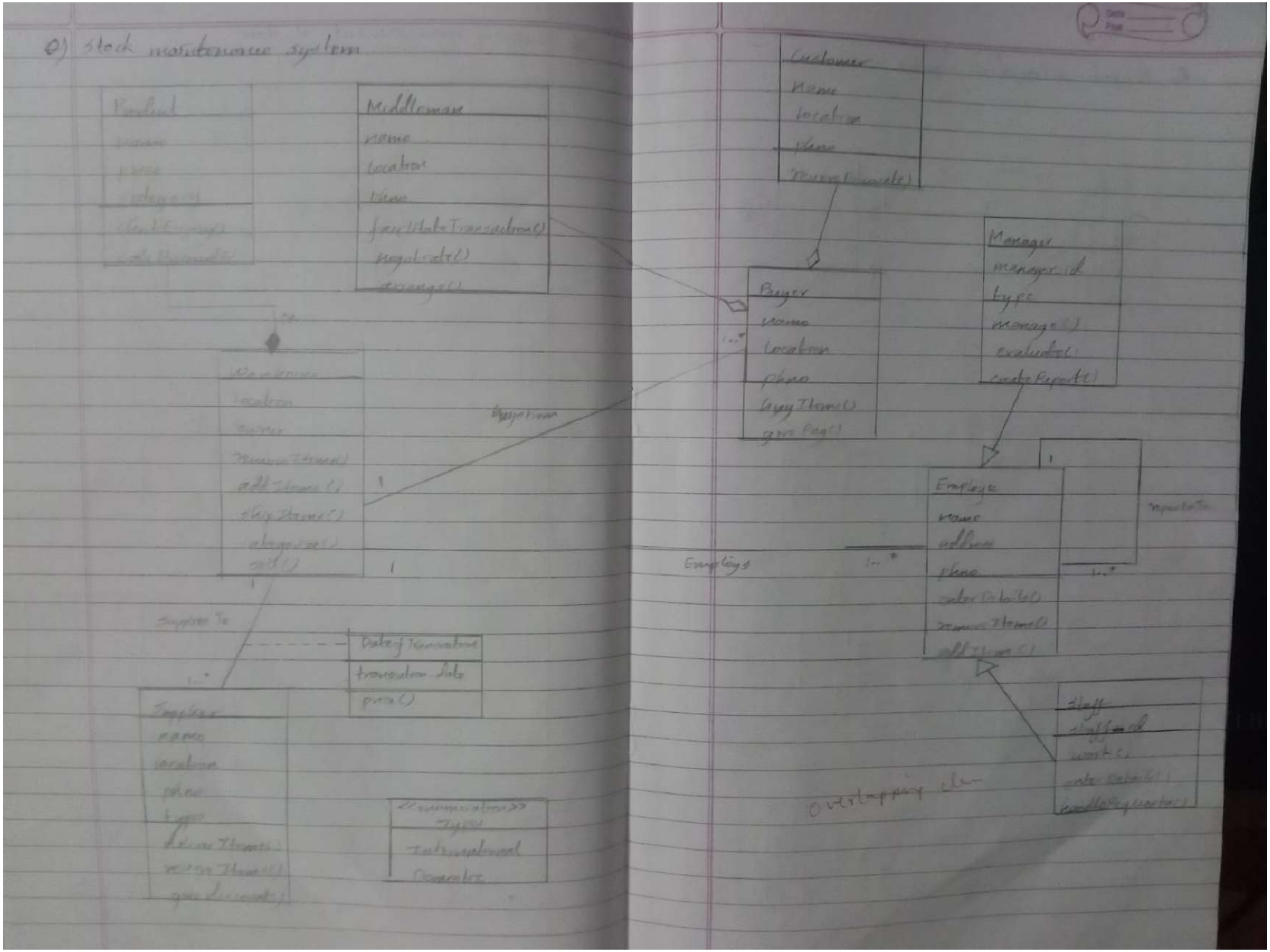
Class diagram represents a system for managing a supply chain or retail business, showcasing the entities involved, their attributes, behaviors, and relationships. It includes classes like Product, Warehouse, Supplier, Middleman, Buyer, Customer, Employee, Manager, and Staff.

The Product class represents items categorized as Food, Stationary, Clothes, Fertilizers, or Chemicals, with methods to check expiry and calculate discounts. Products are stored and managed in the Warehouse, which handles inventory operations like adding, removing, and categorizing items. Suppliers provide products to the warehouse, while Buyers and Customers purchase them, facilitated by a Middleman who negotiates prices and arranges deliveries.

The Employee class represents workers managing warehouse or customer-related tasks, with roles divided into Staff (handling customer requests) and Managers (supervising staff and generating reports). The Date Of Transaction class records transaction details like stock and pricing.

- Warehouses being supplied by Suppliers and serving Buyers.
- Employees reporting to Managers and working within the warehouse or customer service.

This system models the complex interactions between entities in a supply chain, ensuring efficient inventory and transaction management.


Q) stock maintenance system
Middleman
name
location
phno
Customer
Name
location
phno
Buyer
name
location
phno
Manager
manager id
type
Employee
name
address
phno
Supplier
name
location
phno
type
Staff
Overlapping class

b. Advanced State Diagram:

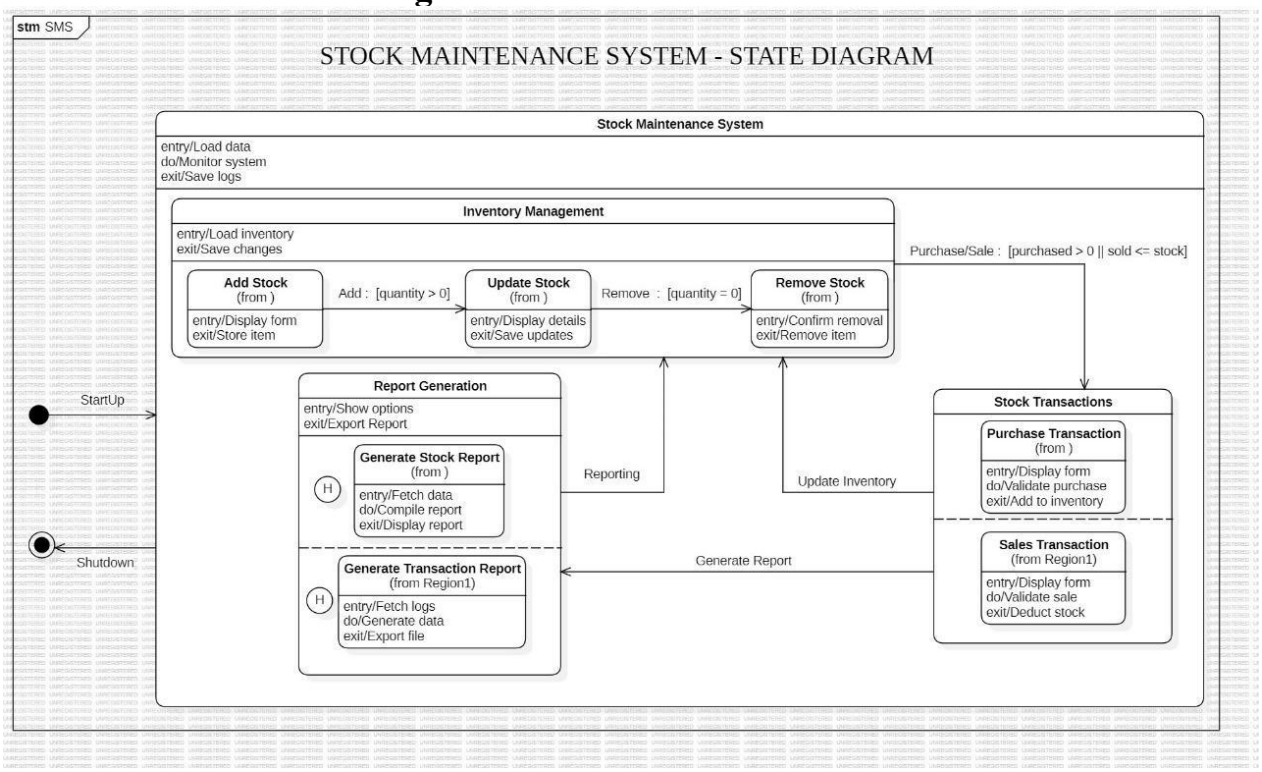
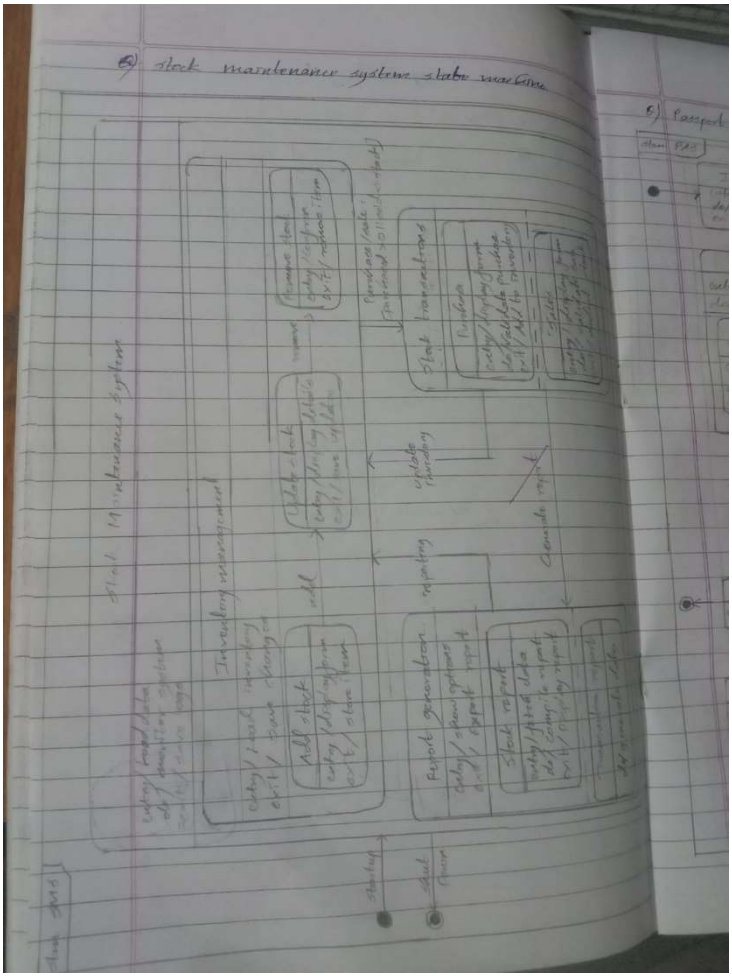


Fig 4.2:



c. Use Case Diagram:

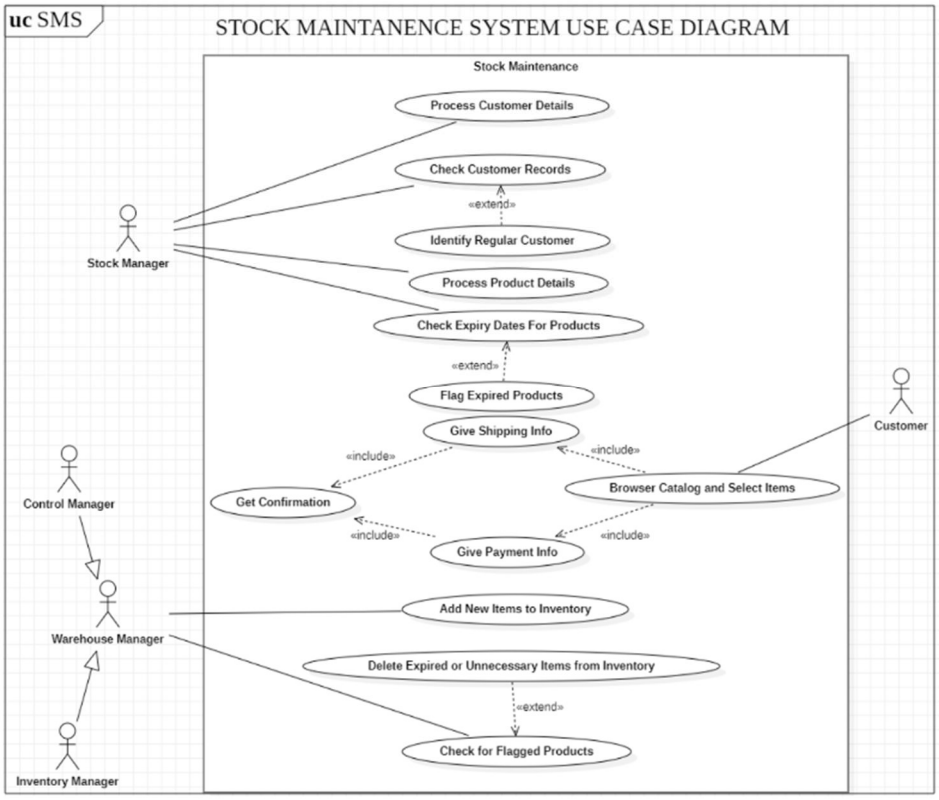
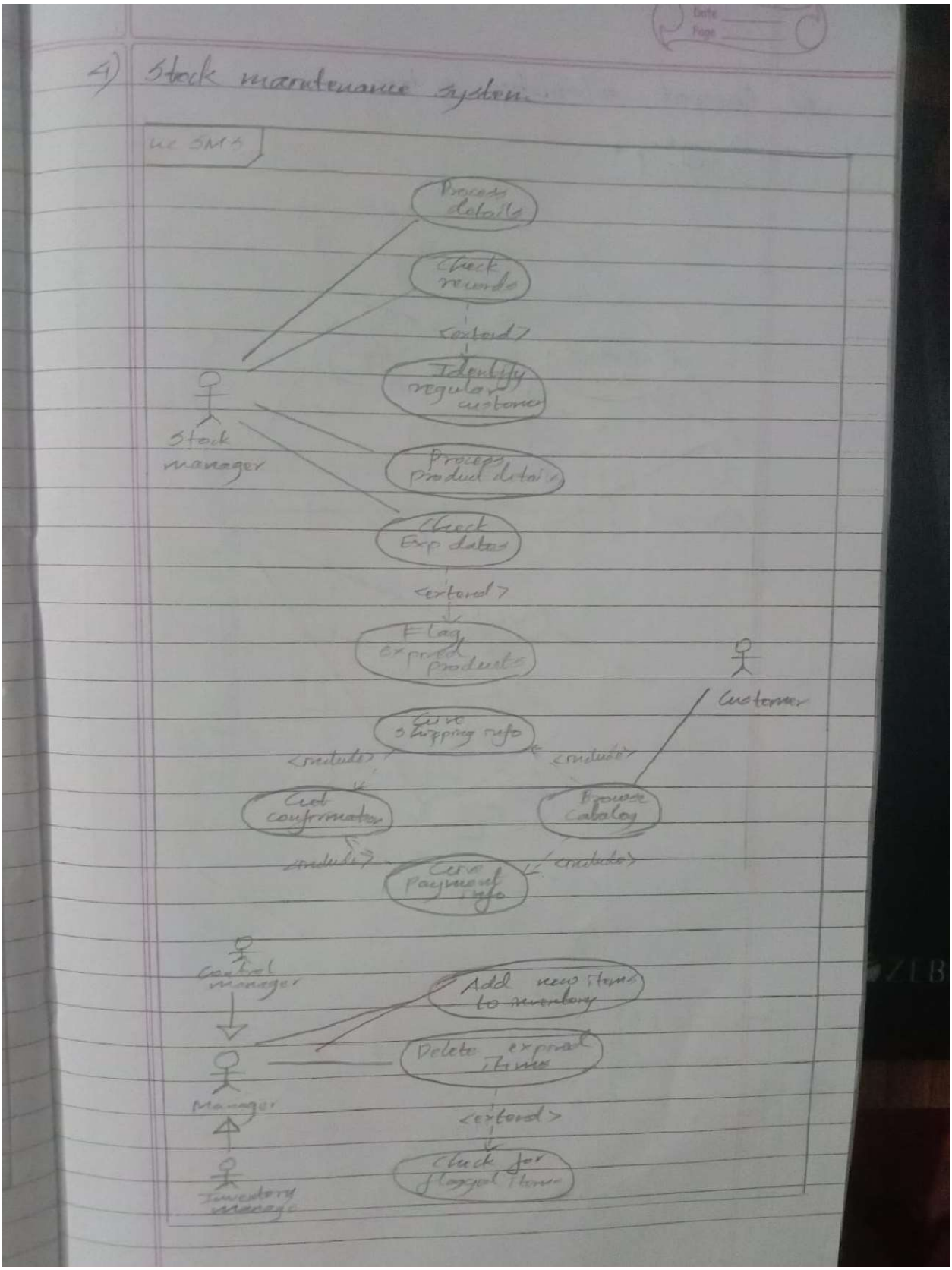


Fig 4.3:

Use case diagram represents a Stock Maintenance System, highlighting the interactions between key actors and system processes:

1. Actors:
    - Stock Manager: Handles customer and product details, checks expiry dates, and flags expired products.
    - Warehouse Manager: Adds new items to inventory.
    - Inventory Manager: Deletes expired or flagged items from inventory.
    - Customer: Browses the catalogue, selects items, and provides payment and shipping details.
2. Key Use Cases:
    - Stock Maintenance: Includes managing customer records, identifying regular customers, and processing product details.
    - Inventory Management: Adding new items and removing expired or unnecessary products.
3. Relationships:
    - Include: Mandatory actions, such as payment info during order processing.
    - Extend: Optional tasks, like flagging expired products during expiry checks.


4) Stock maintenance system
uc SMS
Process details
Check records
<extend>
Identify regular customer
Stock manager
Process product details
Check Exp dates
<extend>
Flag expired products
Customer
Give shipping info
<include>
<include>
Cart confirmation
Browse Catalog
<include>
<include>
Give Payment info
Central manager
Add new items to inventory
Delete expired items
Manager
<extend>
Check for illegal items
Inventory manager

d. Sequence Diagram:

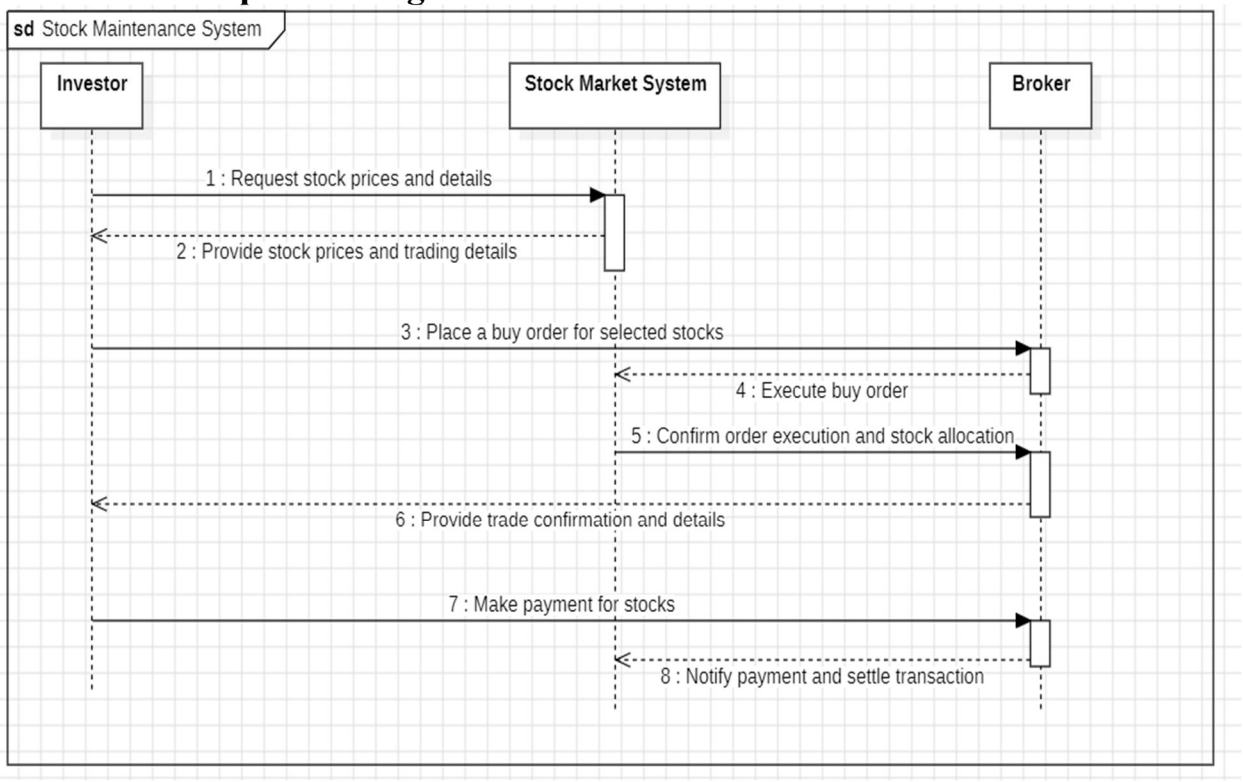
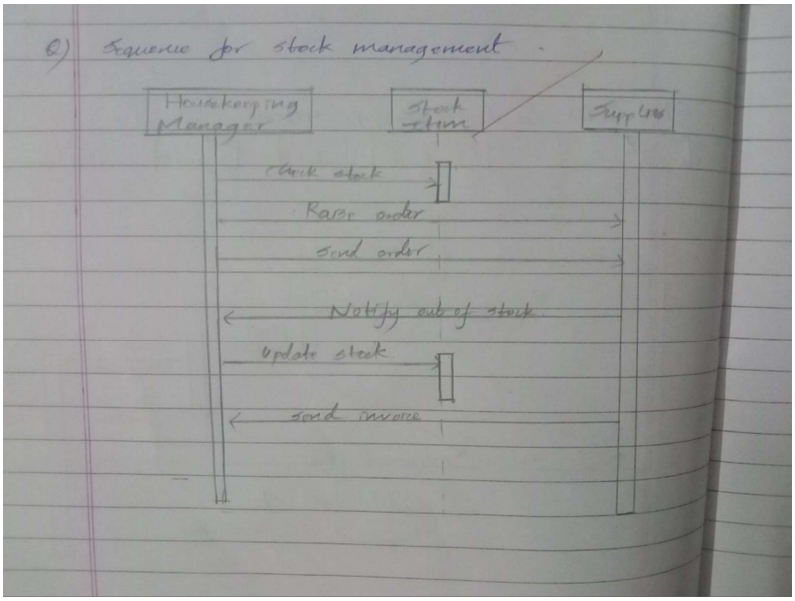


Fig 4.4:



e. Activity Diagram:

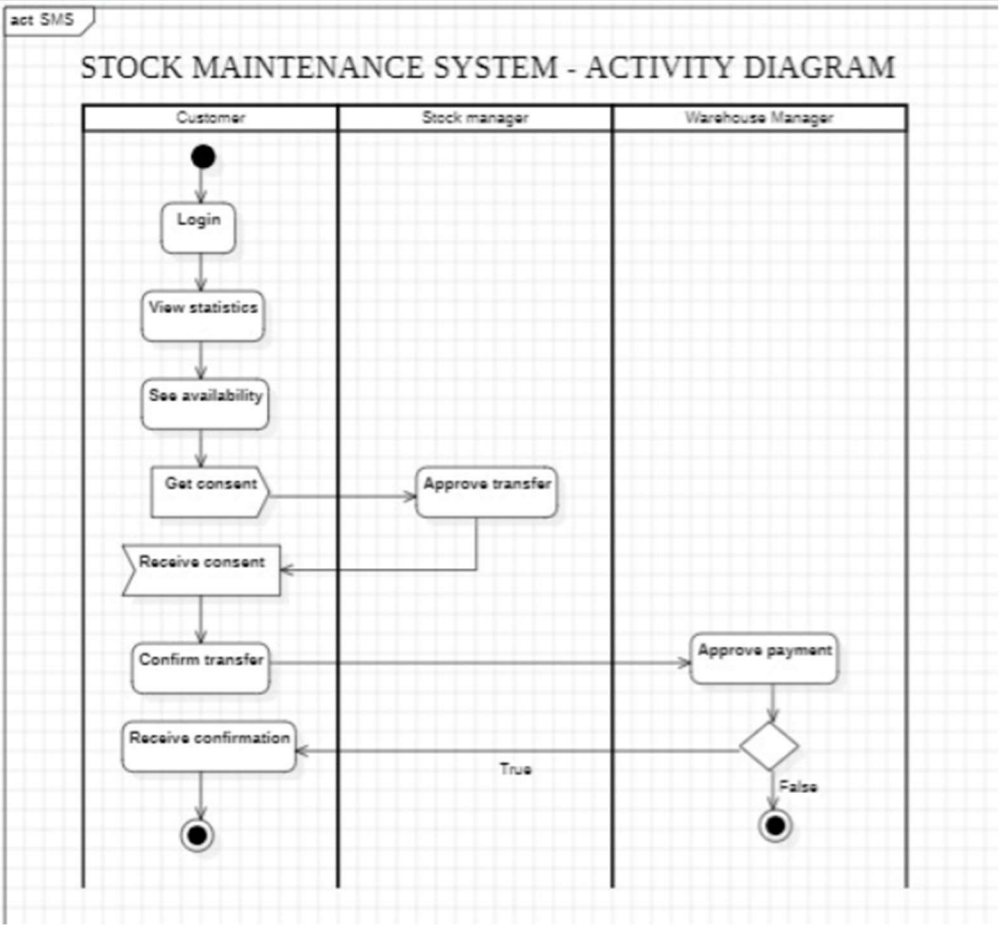
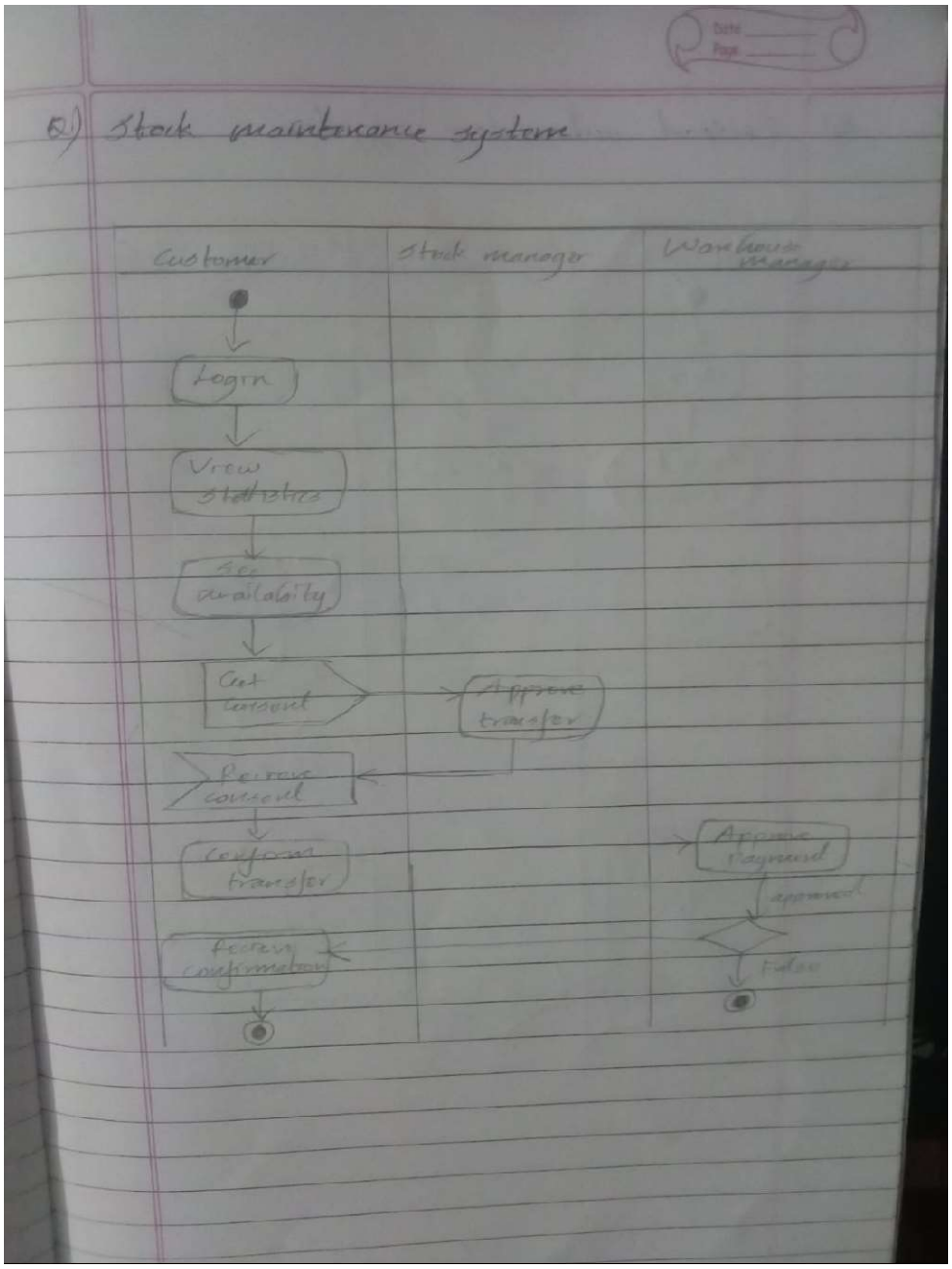
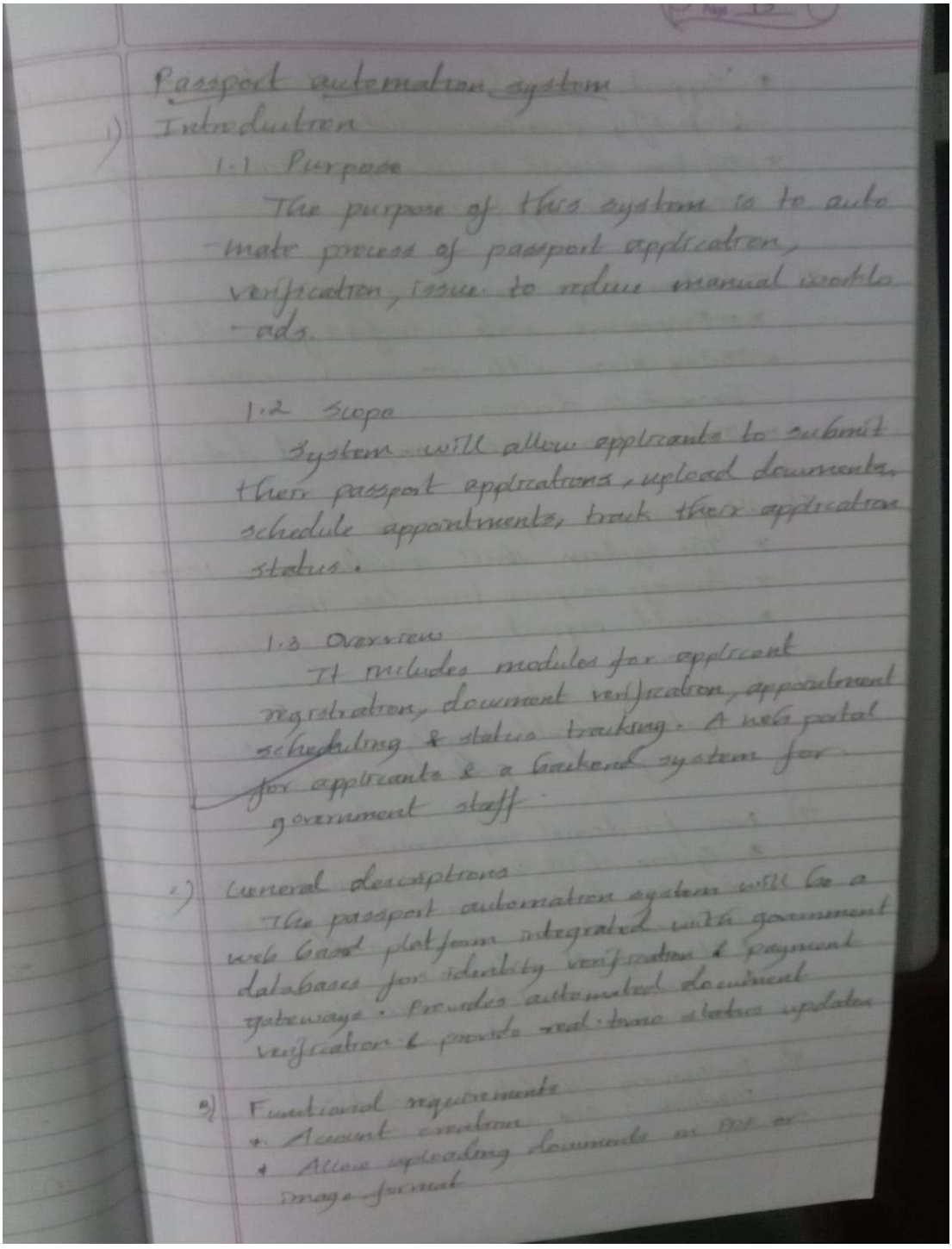


Fig 4.5:


Stock maintenance system
Customer
Stock manager
Warehouse manager
Login
View statistics
See availability
Get consent
Approve transfer
Receive consent
Confirm transfer
Receive confirmation
Approve payment
approved
Pulse

5. Passport Automation

a. SRS Document:



Passport automation system

1) Introduction

1.1 Purpose

The purpose of this system is to automate process of passport application, verification, issue. to reduce manual workloads.

1.2 Scope

System will allow applicants to submit their passport applications, upload documents, schedule appointments, track their application status.

1.3 Overview

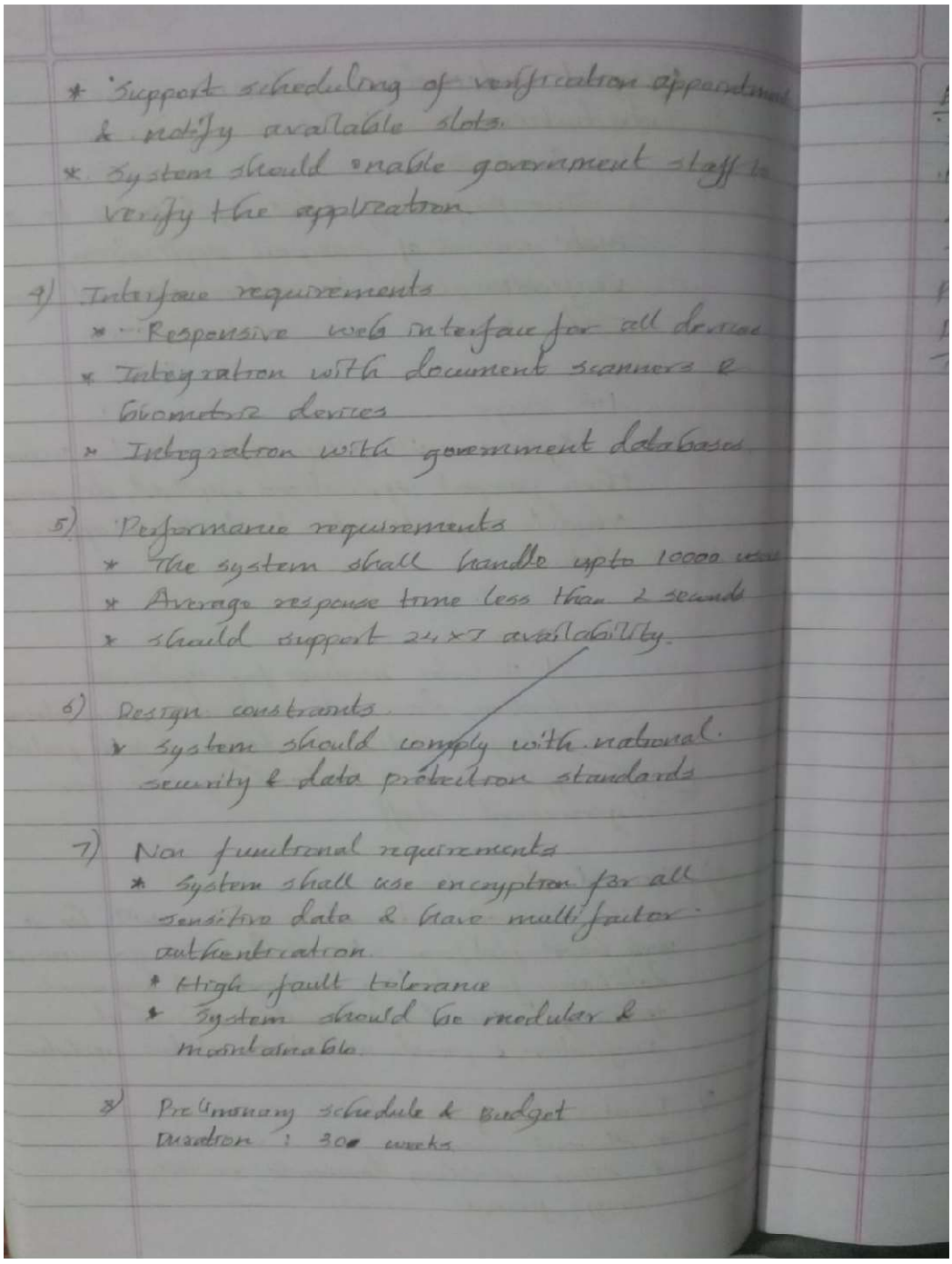
It includes modules for applicant registration, document verification, appointment scheduling & status tracking. A web portal for applicants & a backend system for government staff.

2) General descriptions

The passport automation system will be a web based platform integrated with government databases for identity verification & payment gateways. Provides automated document verification & provide real-time status updates

3) Functional requirements

* Account creation
* Allow uploading documents in PDF or image format



* Support scheduling of verification appointment & notify available slots.
* System should enable government staff to verify the application.

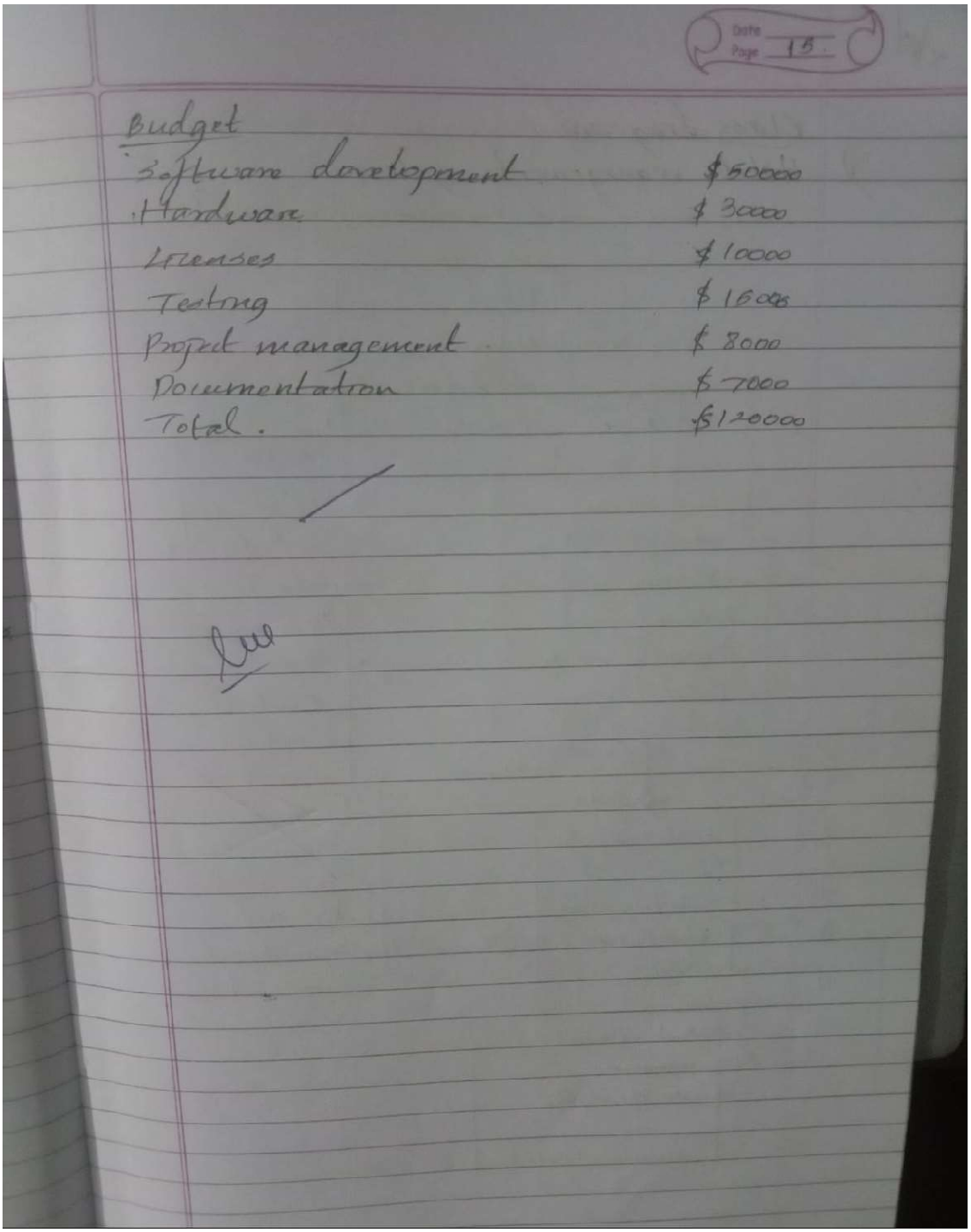
4) Interface requirements
* Responsive web interface for all devices
* Integration with document scanners & biometric devices
* Integration with government databases

5) Performance requirements
* The system shall handle upto 10000 users
* Average response time less than 2 seconds
* Should support 24×7 availability.

6) Design constraints
* System should comply with national security & data protection standards

7) Non functional requirements
* System shall use encryption for all sensitive data & have multi factor authentication.
* High fault tolerance
* System should be modular & maintainable.

8) Preliminary schedule & Budget
Duration : 300 weeks


Date
Page 15
Budget
Software development $50000
Hardware $30000
Licenses $10000
Testing $16000
Project management $8000
Documentation $7000
Total. $120000

b. Advanced Class Diagram:

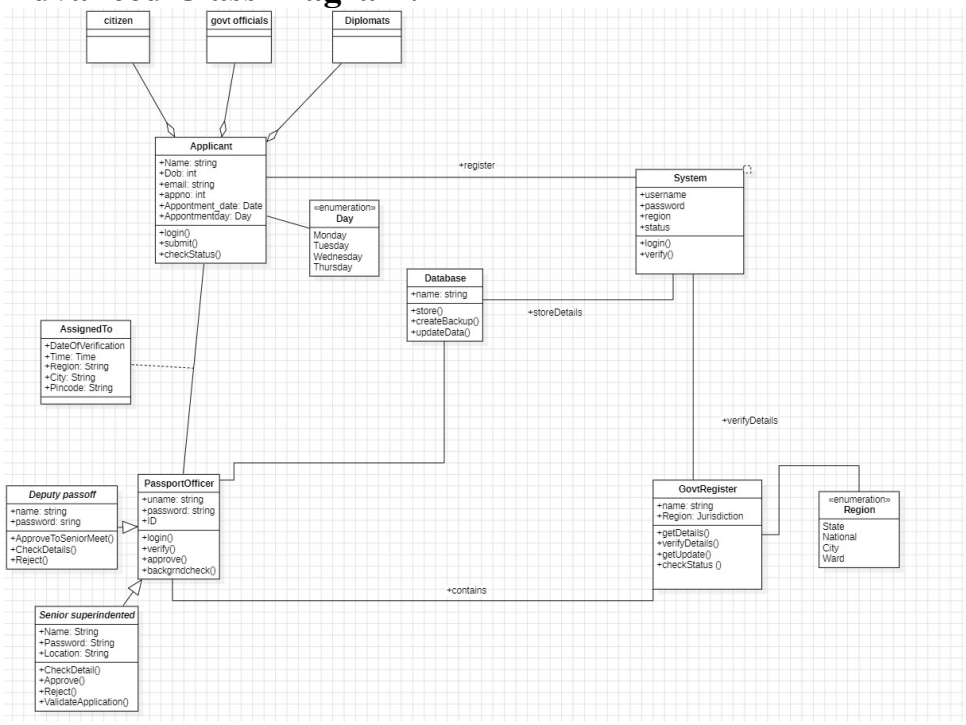
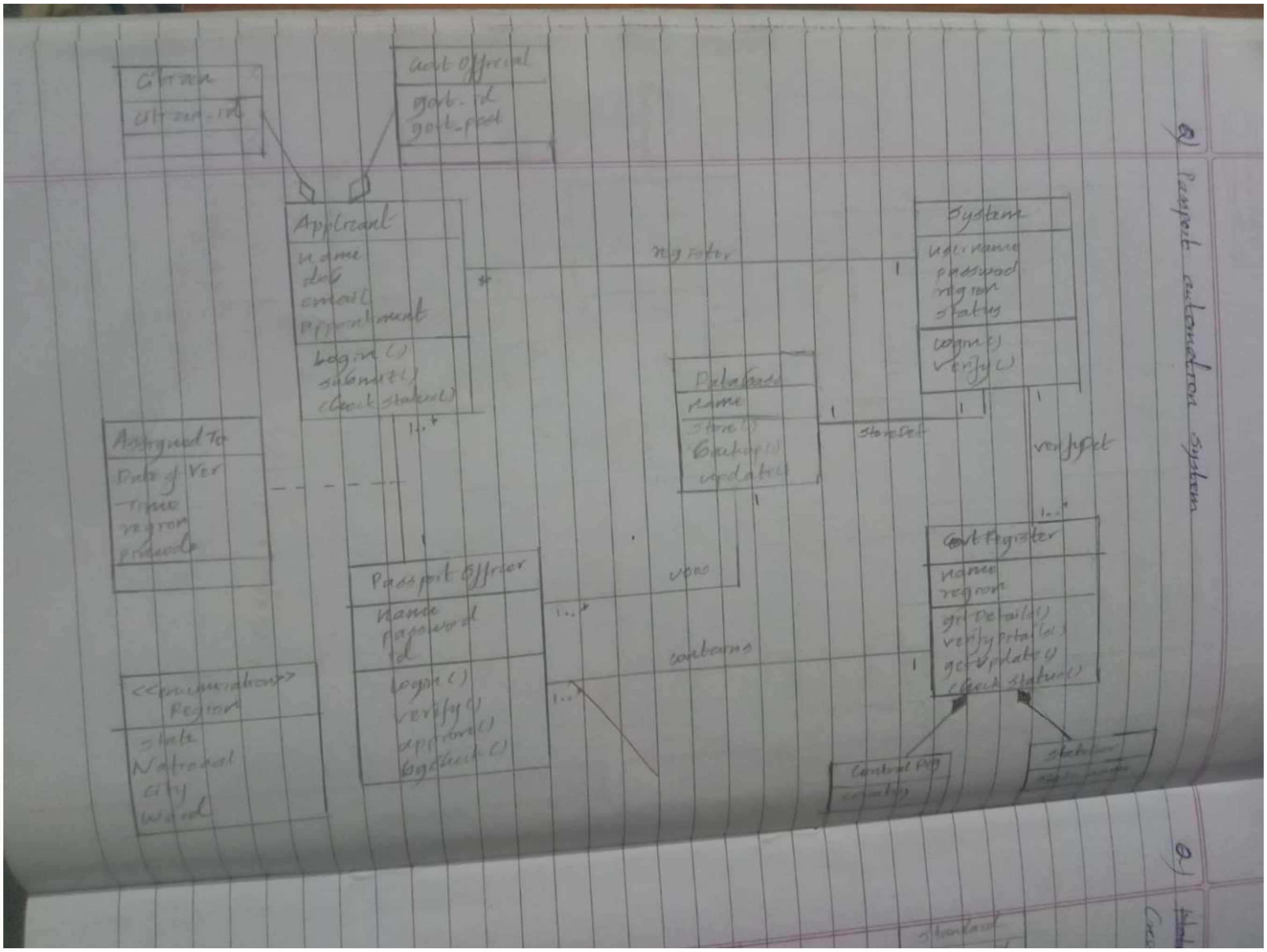


Fig 5.1

Passport management system that facilitates the process of applying for and managing passport applications. The key actors include applicants such as citizens, government officials, and diplomats, who interact with the system to submit their applications, book appointments, and track their status. Applicants provide essential details like their name, date of birth, and appointment information and can log in to the system to check updates on their application.
The system itself handles the core processes, such as user registration, login, and verification of details. All the information is stored and managed in a central database, which ensures that application data is securely stored, backed up, and updated as needed. Applications are assigned to specific Passport Officers based on the region, appointment schedule, and other factors. These officers are responsible for verifying documents, approving or rejecting applications, and updating the system with their decisions.
The process is overseen by Senior Superintendents, who validate and approve final decisions to ensure accuracy and compliance. Additionally, a Cover Register tracks applications based on their jurisdiction, categorizing them by state, city, or ward, and ensuring they are routed to the correct authorities.
This system organizes tasks efficiently by linking the roles of applicants, officials, and system components. It simplifies the workflow, ensures data security, and provides transparency for applicants to monitor their application progress.


Q) Passport automation system
Citizen
Govt Official
Applicant
name
dob
email
login()
submit()
check status()
register
System
username
password
region
status
login()
verify()
Database
name
store()
update()
storeDet
verifyDet
Assigned To
Date of Ver
Time
region
pincode
Passport Officer
Name
password
id
login()
verify()
approval()
views
contains
Govt Register
name
region
getDetails()
verifyDetails()
getUpdate()
check status()
Central Reg
<<enumeration>>
Region
State
National
city
ward

c. Advanced State Diagram:

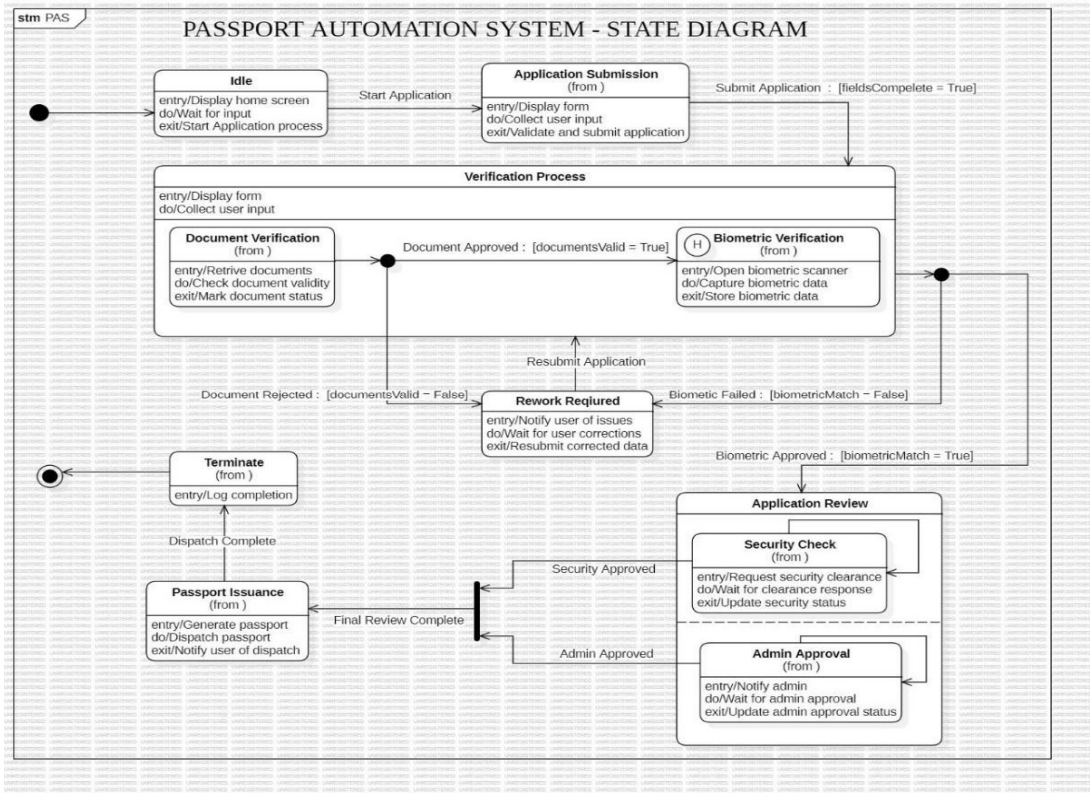
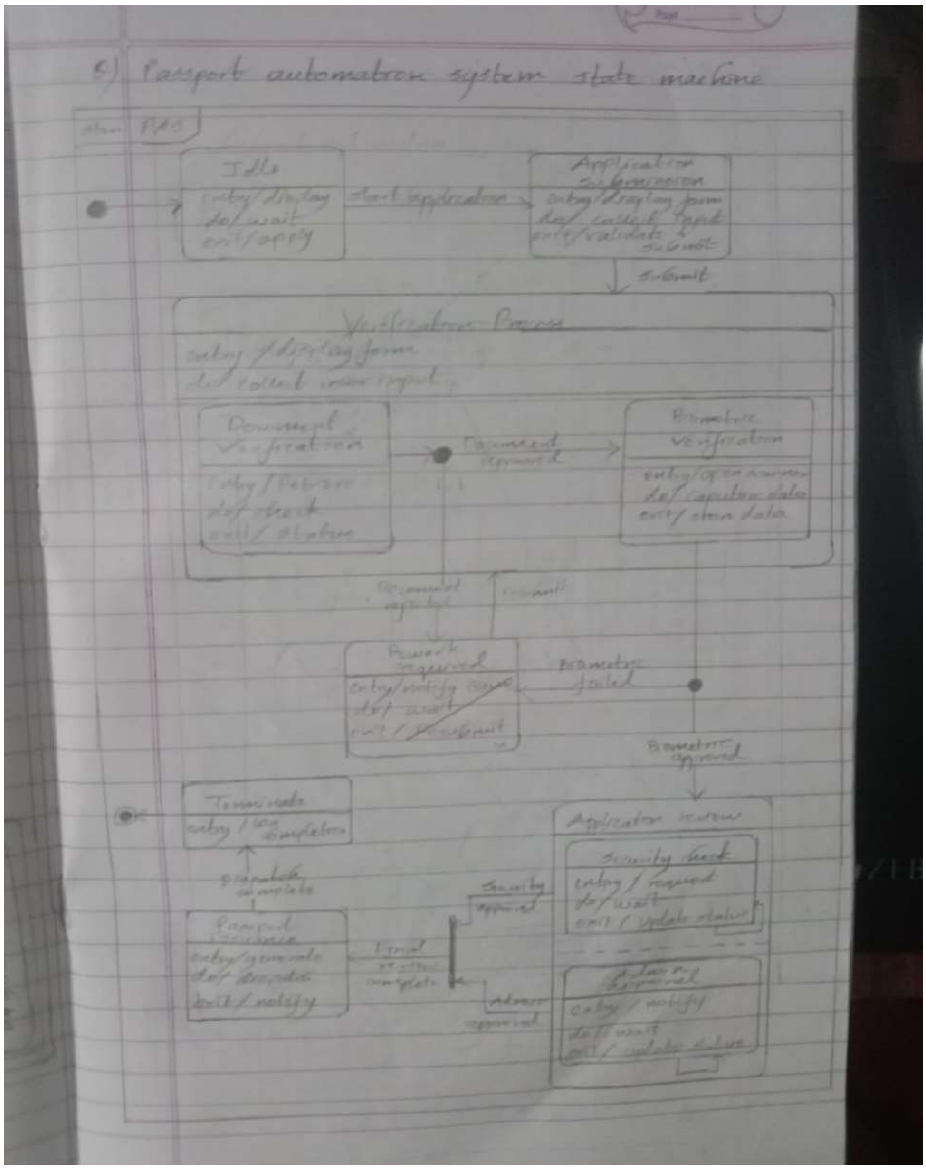


Fig 5.2


5) Passport automation system state machine
Idle
Application submission
Verification Process
Document Verification
Biometric verification
Payment approved
Terminate
Application review
Security check
Final approval

d. Use Case Diagram:

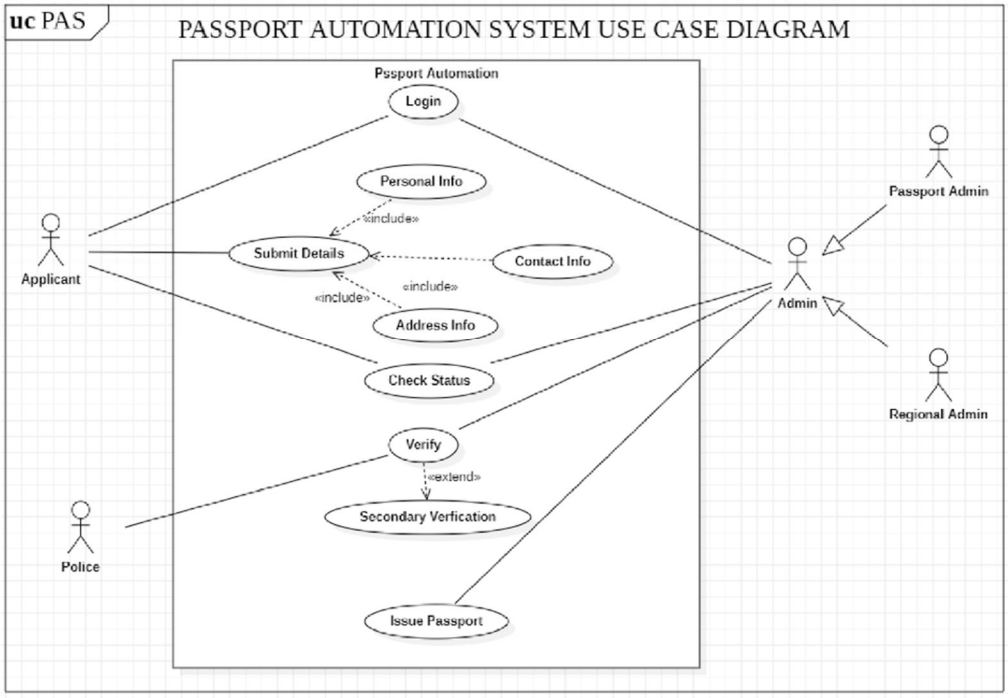
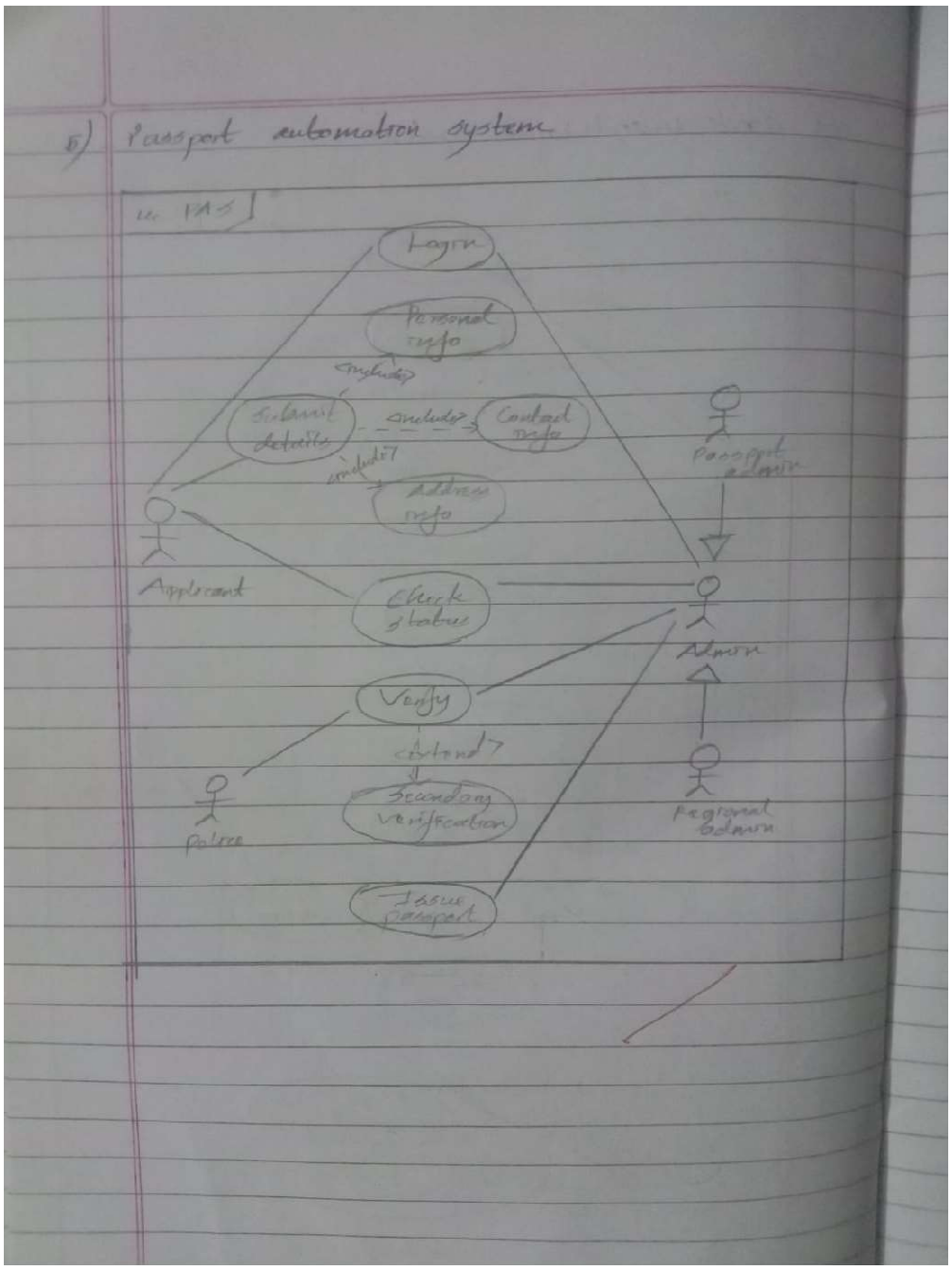


Fig 5.3


5) Passport automation system
Use CASE
Login
Personal info
Submit details
Contact info
Address info
Applicant
Check status
Verify
Police
Passport Admin
Admin
Regional Admin
Issue passport

e. Sequence Diagram:

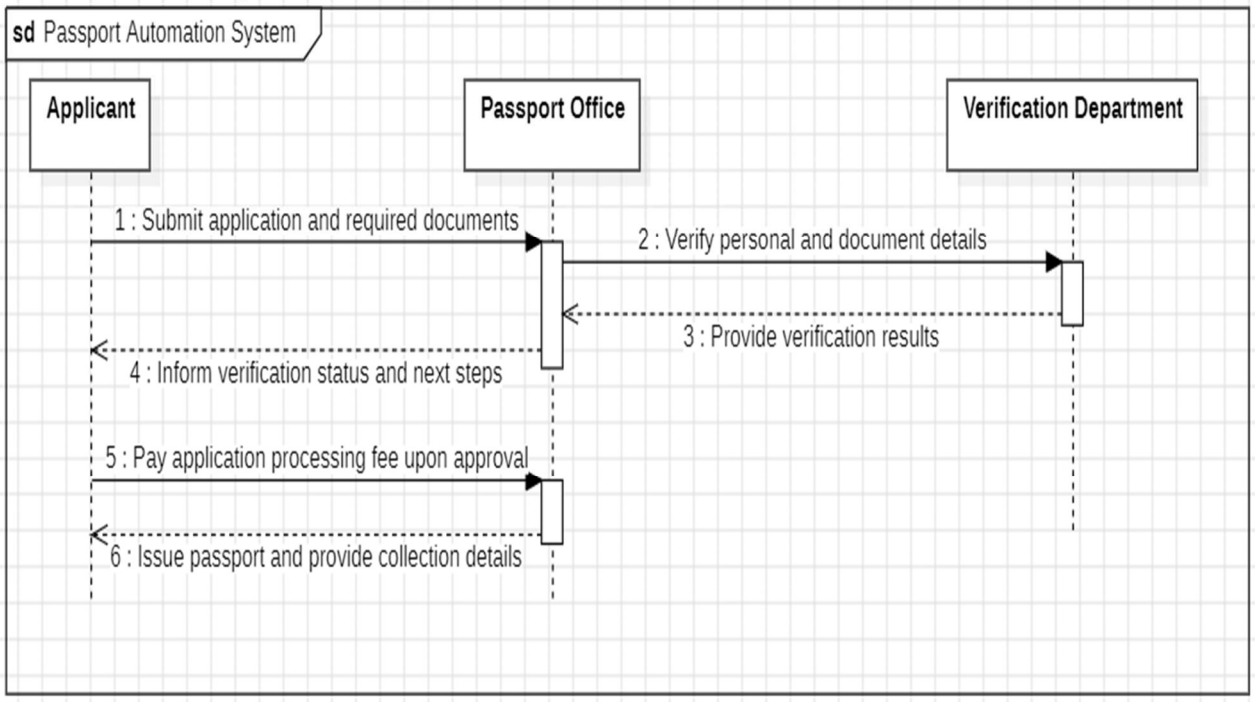
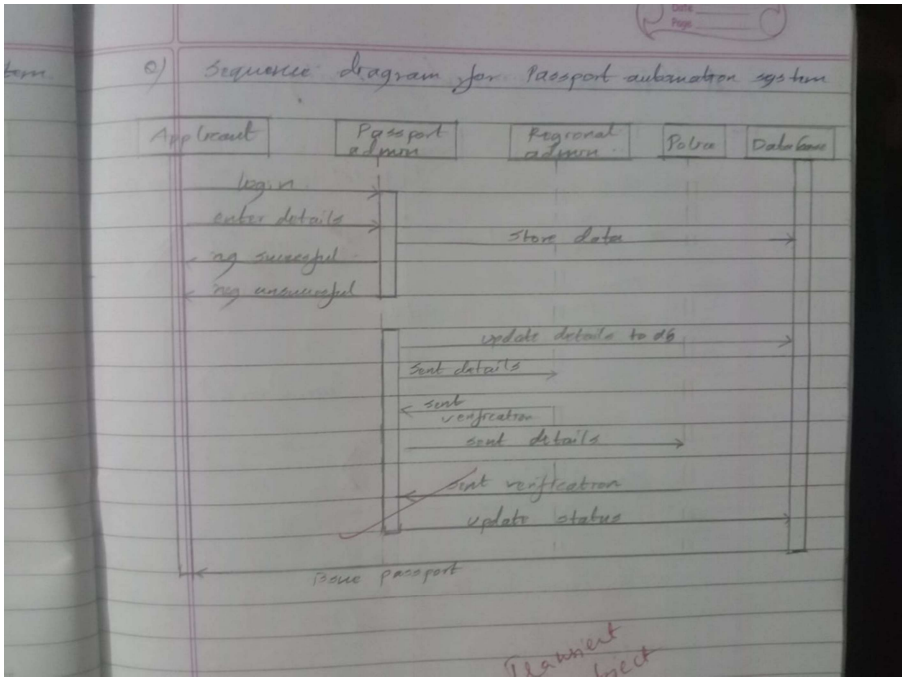


Fig 5.4



f. Activity Diagram:

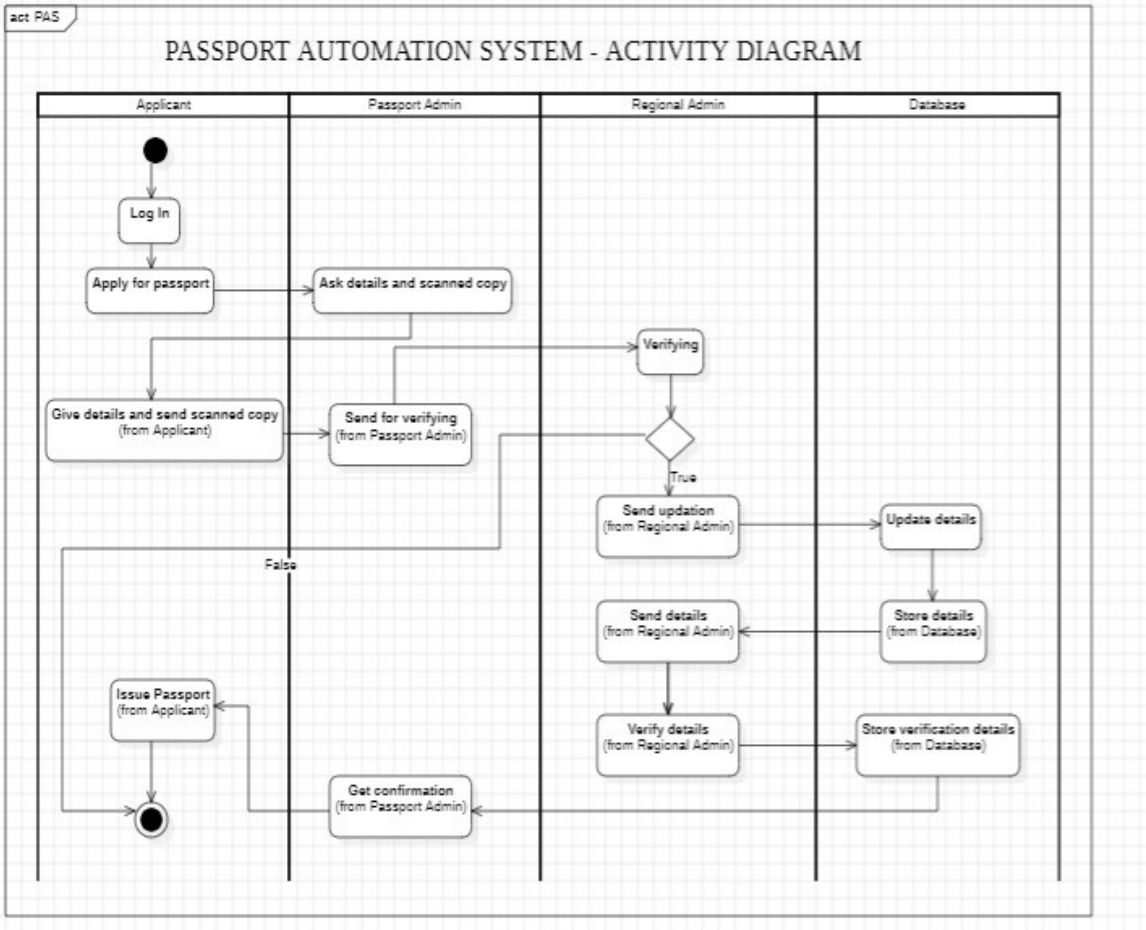
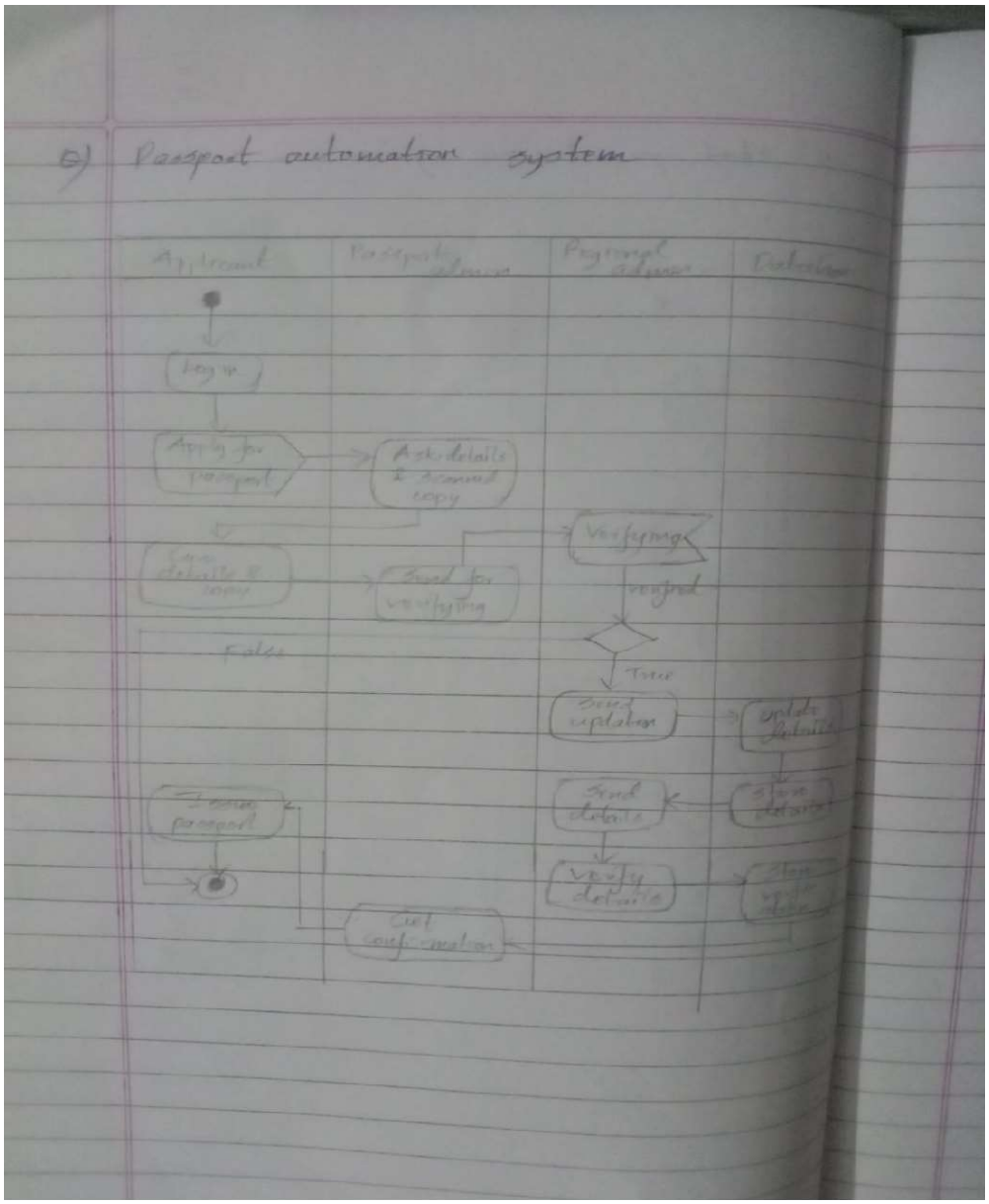


Fig 5.5


6) Passport automation system
Applicant
Passport officer
Police verification
Database
Login
Apply for passport
Ask details & scanned copy
Verifying
Fill details & upload
Send for verification
Accepted
False
True
Send updation
Update details
Pay passport
Send details
Store details
Verify details
Get confirmation