

# Assignment No. 1

- 1) Make a UML diagram (use case diagram, Class diagram, Object Diagram, Activity diagram) and data flow diagram for library manage.

- 1) Use case diagram notations.



actor



use case



System Boundary

use cases

- 1) adding new books
- 2) borrow book
- 3) Return a book
- 4) maintaining member info.
- 5) creating reports for overdue and borrowed books.

## Use Case Diagram Explanation.

- 1) Register new user -

1) The user fills out a reg. form.

2) Librarian issues a library card with an ID card to user.

- 2) Request new book -

1) User searches and requests a book as per their requirement.

2) Requested book is Reserved for user.

- 3) Borrow book -

1) After requesting user borrow book

2) Librarian updates borrow records in db.

#### 4) Renew Book -

- 1) User can renew the book to extend due date.
- 2) Renewal updates the books due data if allow.

#### 5) Return book.

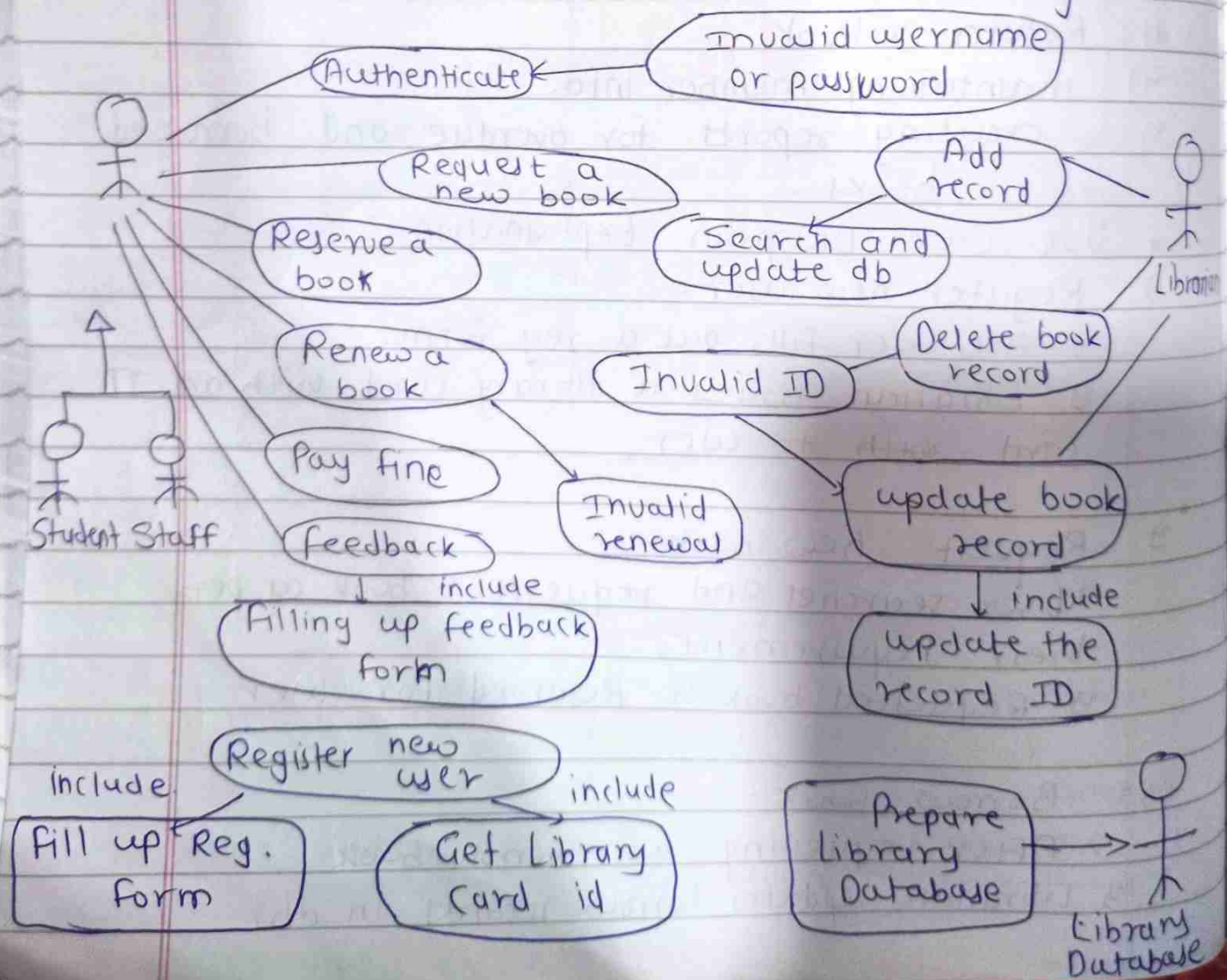
- 1) User returns the borrow book
- 2) Librarian updates return record in system.

#### 6) Pay fine.

- 1) If user does not return or renew book by due date, they pay fine.

#### 7) Provide feedback - User can fill out a feedback form about library services.

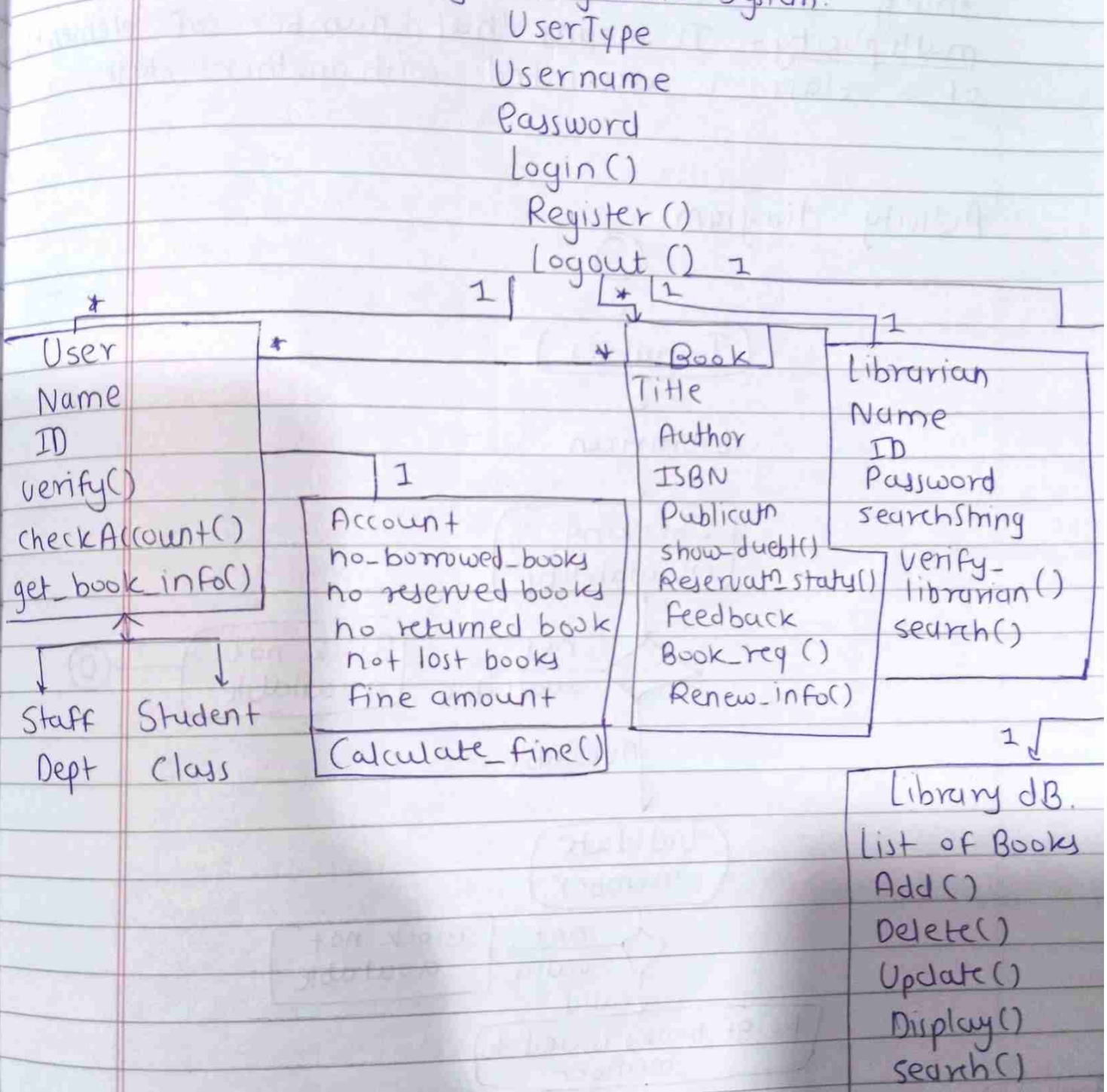
#### 8) manage records - Librarian adds or delete records of users and books when necessary.





## \* Class Diagram.

## Library Management System.



## Problem Statement 2

Design an e-commerce application that supports many payment methods. The system should be easily extensible in future, allowing for addition of additional payment methods without changing existing code. Determine and describe which diagram pattern should be most appropriate for this situation. Provide a thorough solution that incorporates this design patterns including class diagrams & explanations of how it solves the problem.

→ The best design pattern for this problem is strategy pattern that belong to behavioural design pattern in software engineering. It promotes open/closed design where system is open for extension & change existing code when adding new strategies.

4) Payment Incorporates this design patterns

a) Payment Strategy Interface : Defines the common interface that all payment methods will implement.

b) Concrete Payment strategies - There are individual payment methods, such as create Payment

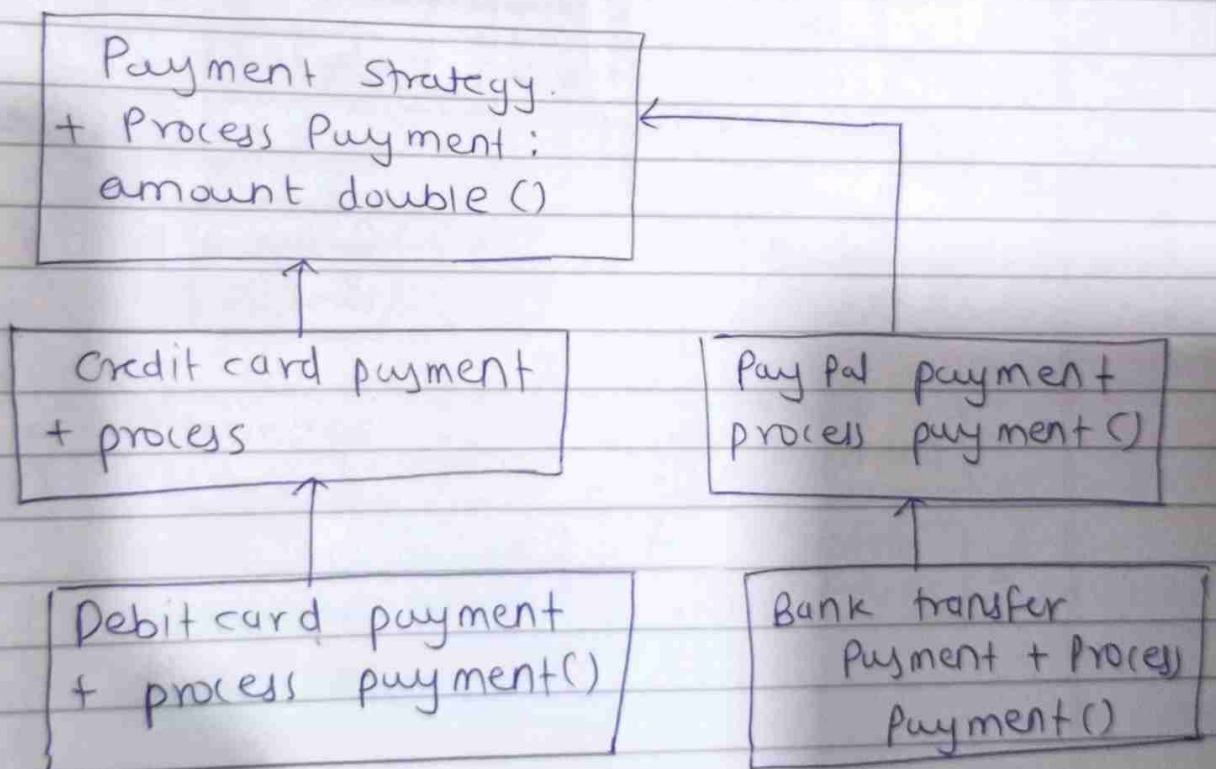
c) Payment Context class - This class maintains a reference to payment strategy object & allow clients to change the payment strategy at runtime



How to solve a problem.

- Scalability** - New payment method (Bitcoin, apple) can be added without modifying existing code by simply creating new class payment strategy.
- Flexibility** - The payment context change user or system depending on scenario.
- Maintainability** - Each payment method can be develop, tested & maintain independently by isolam each in its own ~~glw~~ class without indexing with others.

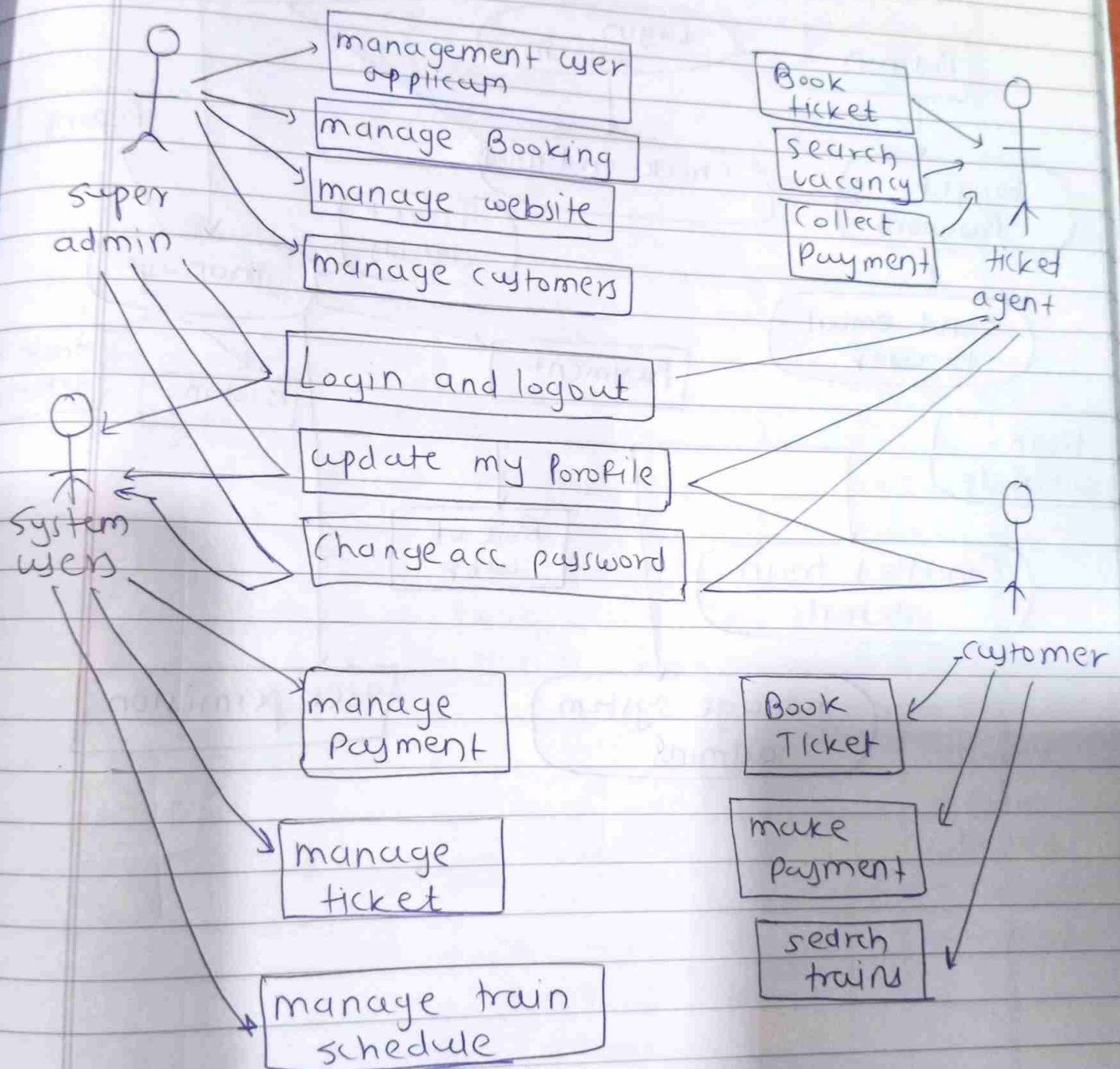
Class Diagram.



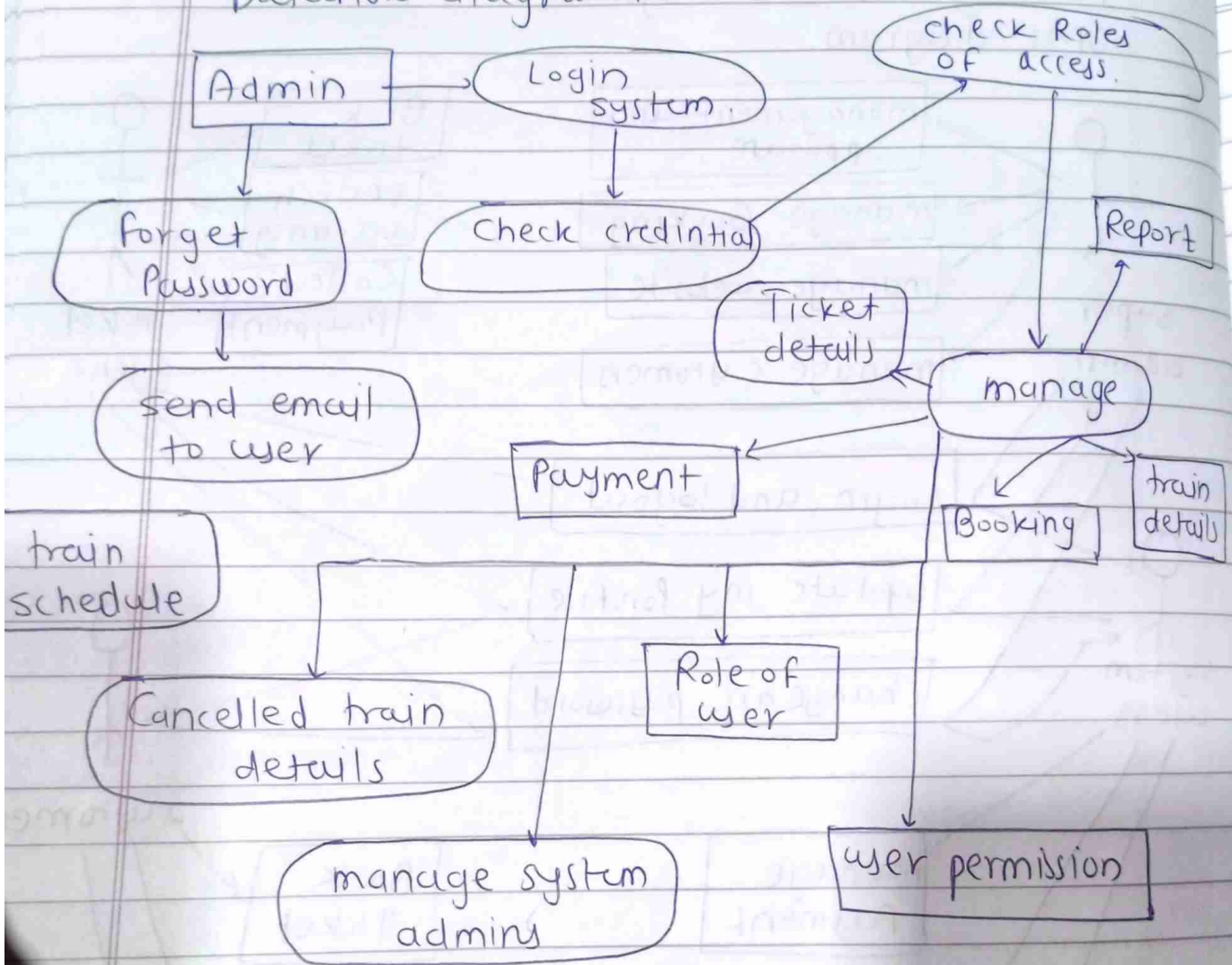
Payment context  
- strategy payment strategy.  
+ Set payment method

# \* Railway Reservation System. UML diagram.

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

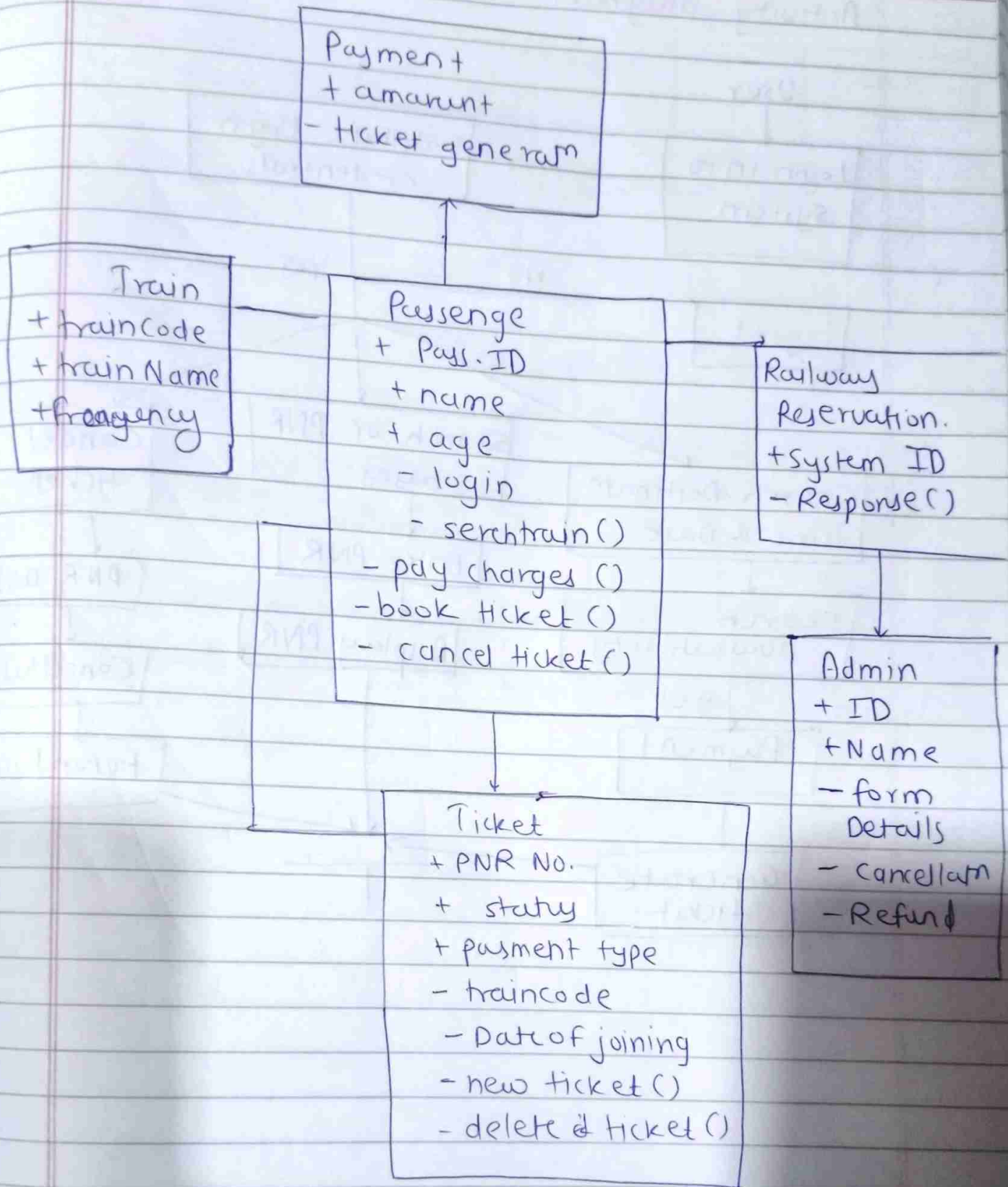


## Dataflow diagram



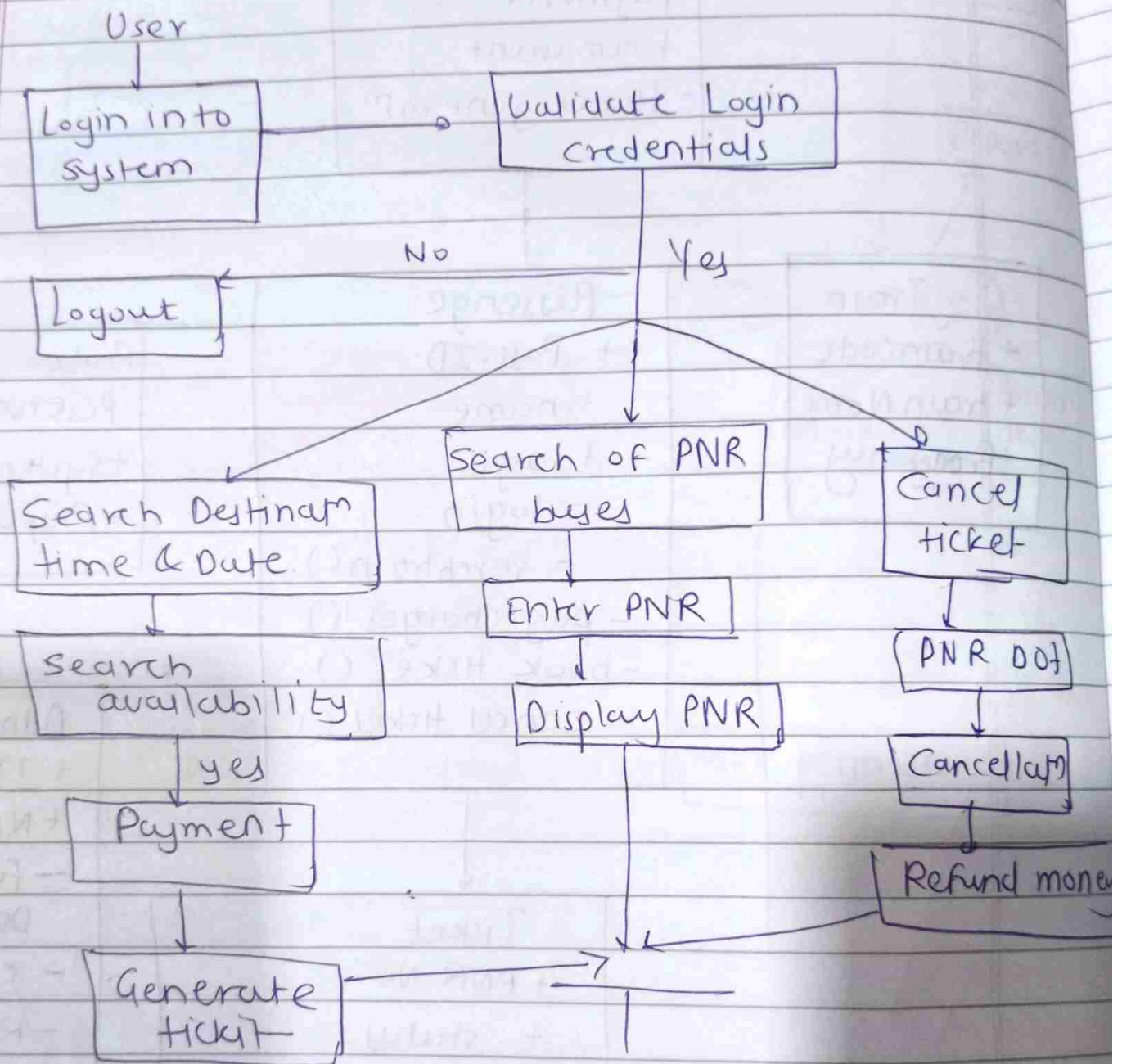


# Class Diagram.

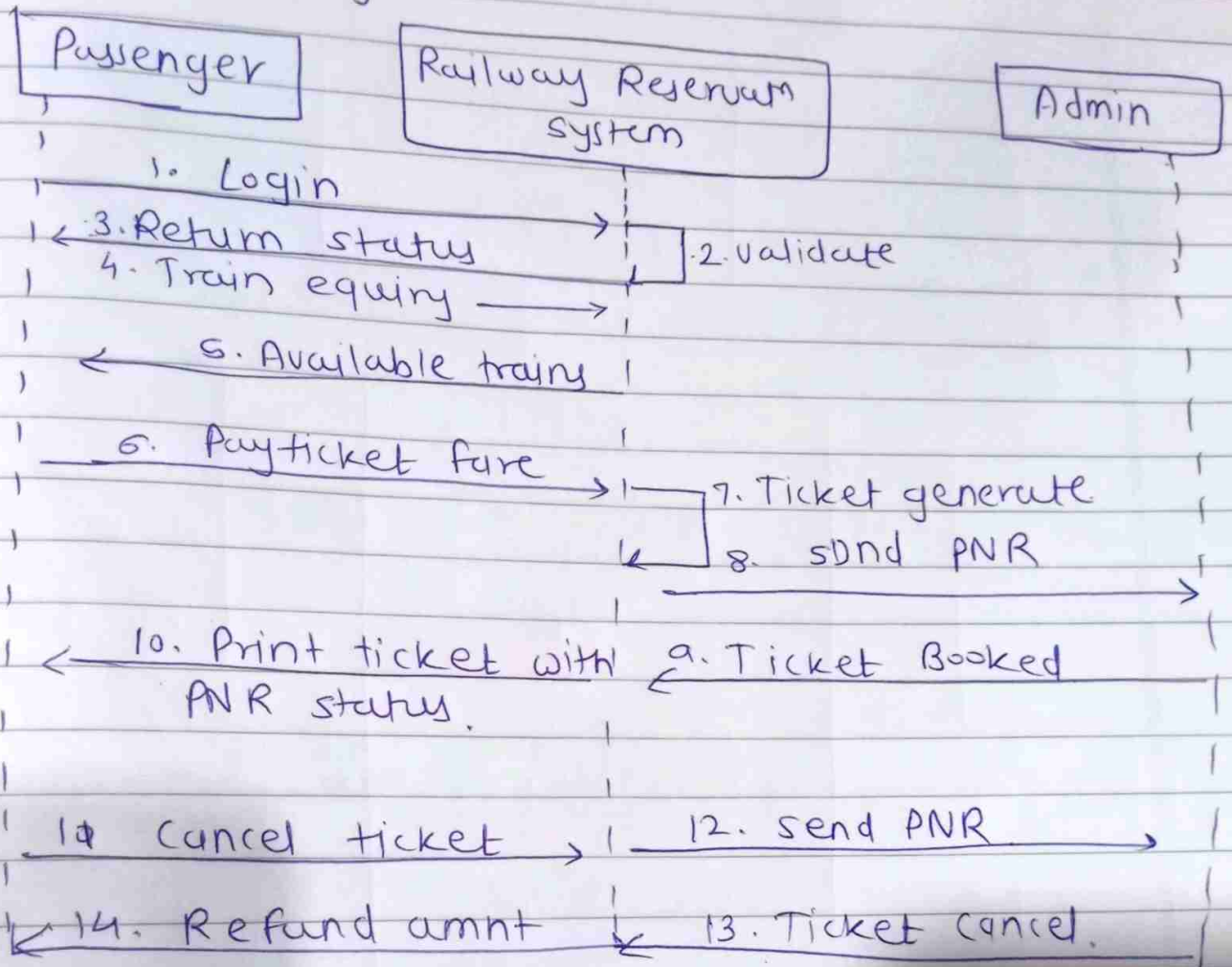




## Activity diagram



# Object Diagram

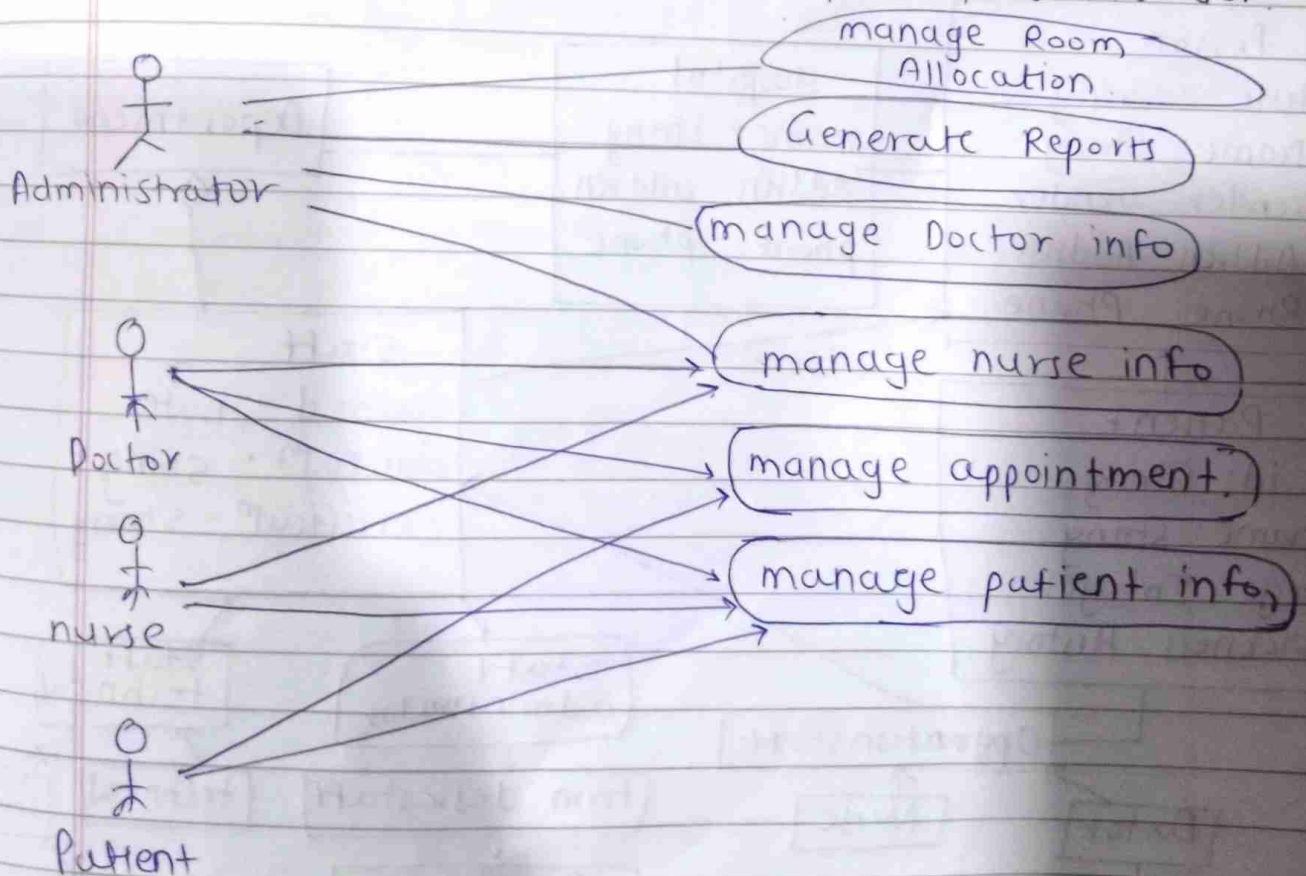


Q. Create Hospital management System that include features like managing patient records scheduling appointments, processing invoicing & payments, managing staff info. and creating reports. Represent these fundamentalities with several UML diagrams and data flow diagram, object diagram, Activity diag. and a data flow diagram (DFD). Provide through illustration & description for each of demonstrate the overall architecture & performance of Hospital management system.

### i) Use Case Diagram

The diagram illustrates the interaction between actors (uses) and system.

Actors - Patient, Doctor, Receptionist, Administrator





use cases -

Patient schedule appointment, view medical history, pay bills.

Doctor - View patient records, schedule appointment, prescribe medications.

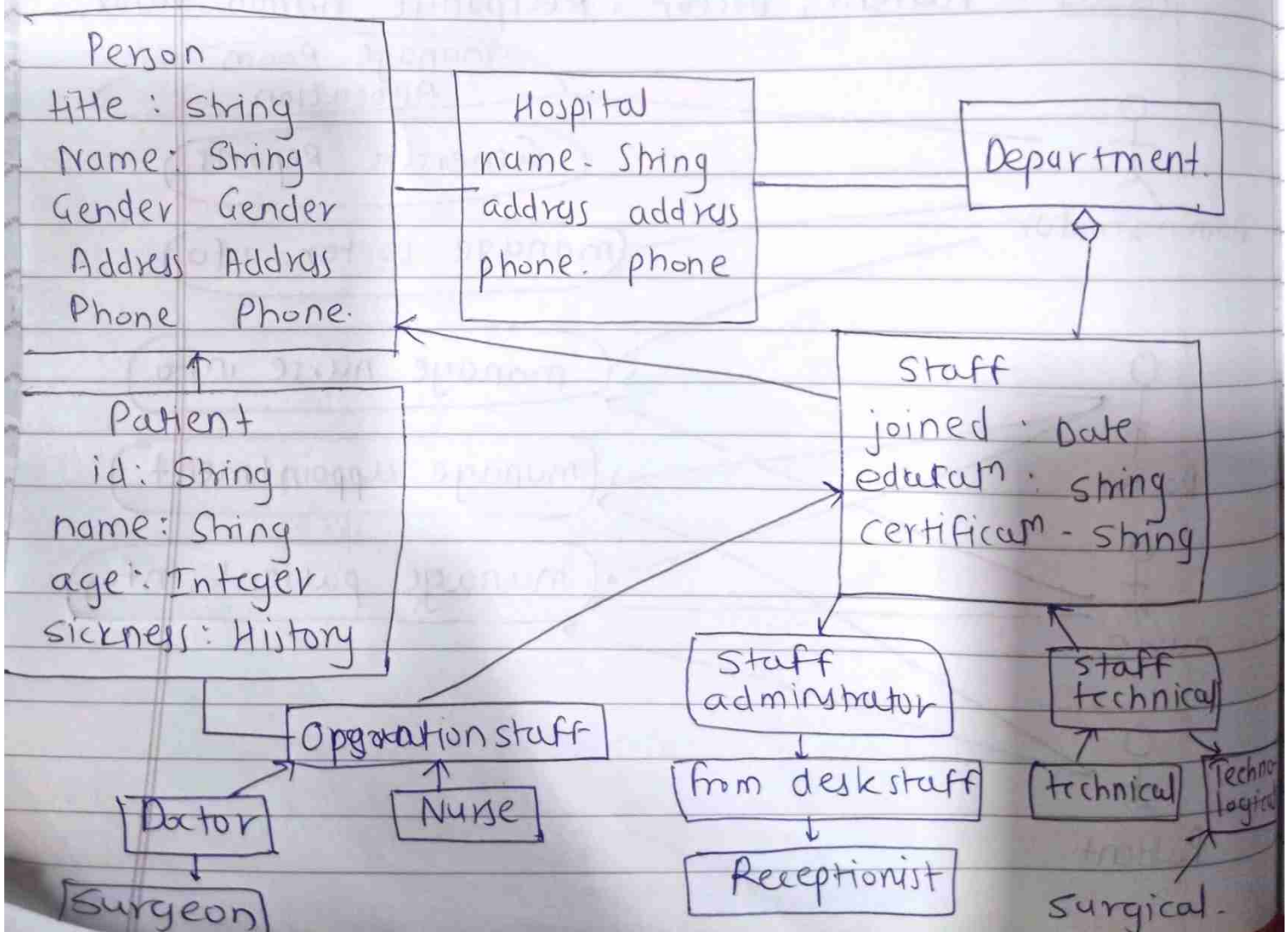
Receptionist schedule appointment, process payments, manage patient records.

Administrator - manage staff info, general reports, update system settings.

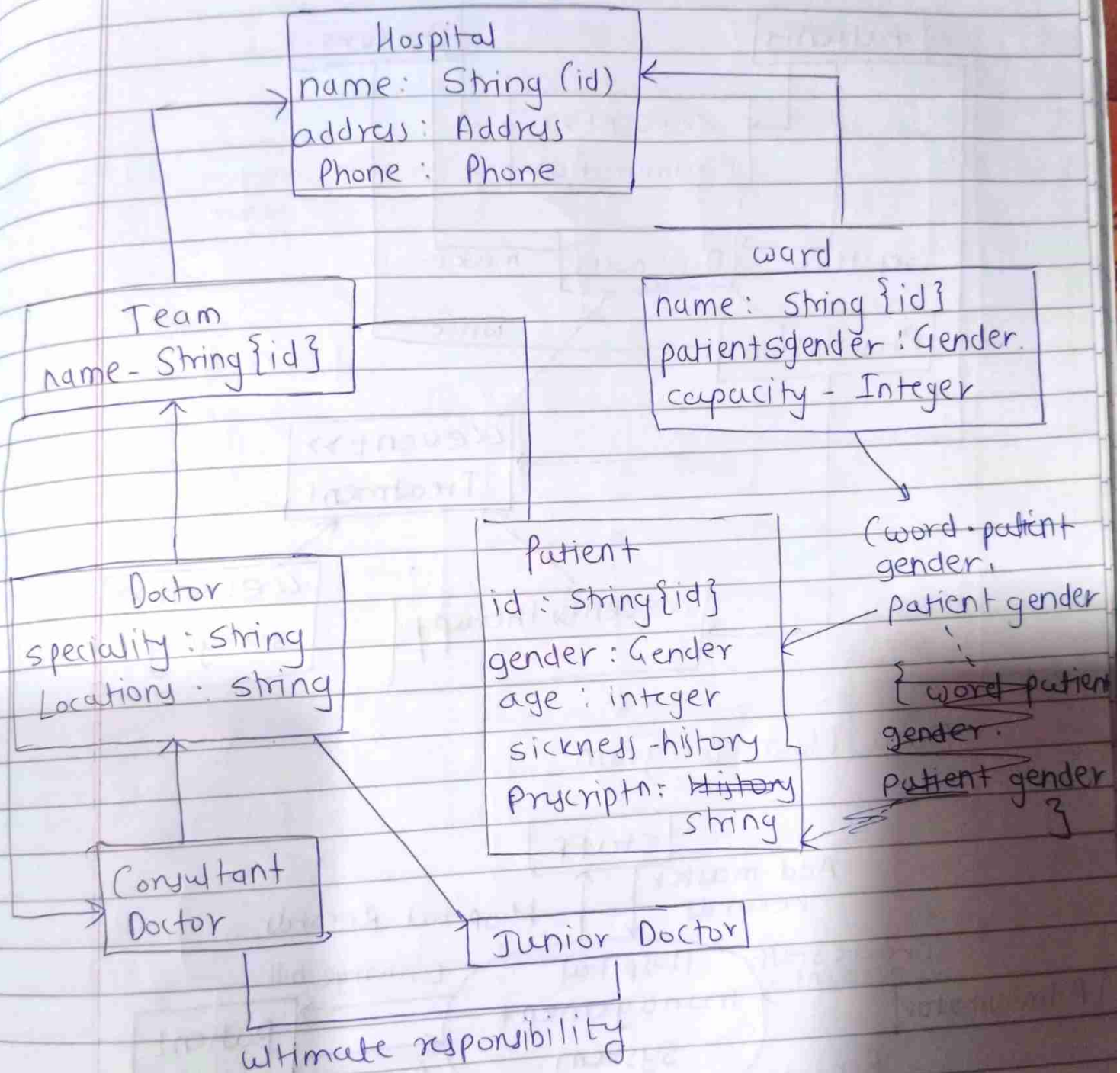
## 25 Class Diagram.

This diagram shows the classes & their relationships in the system.

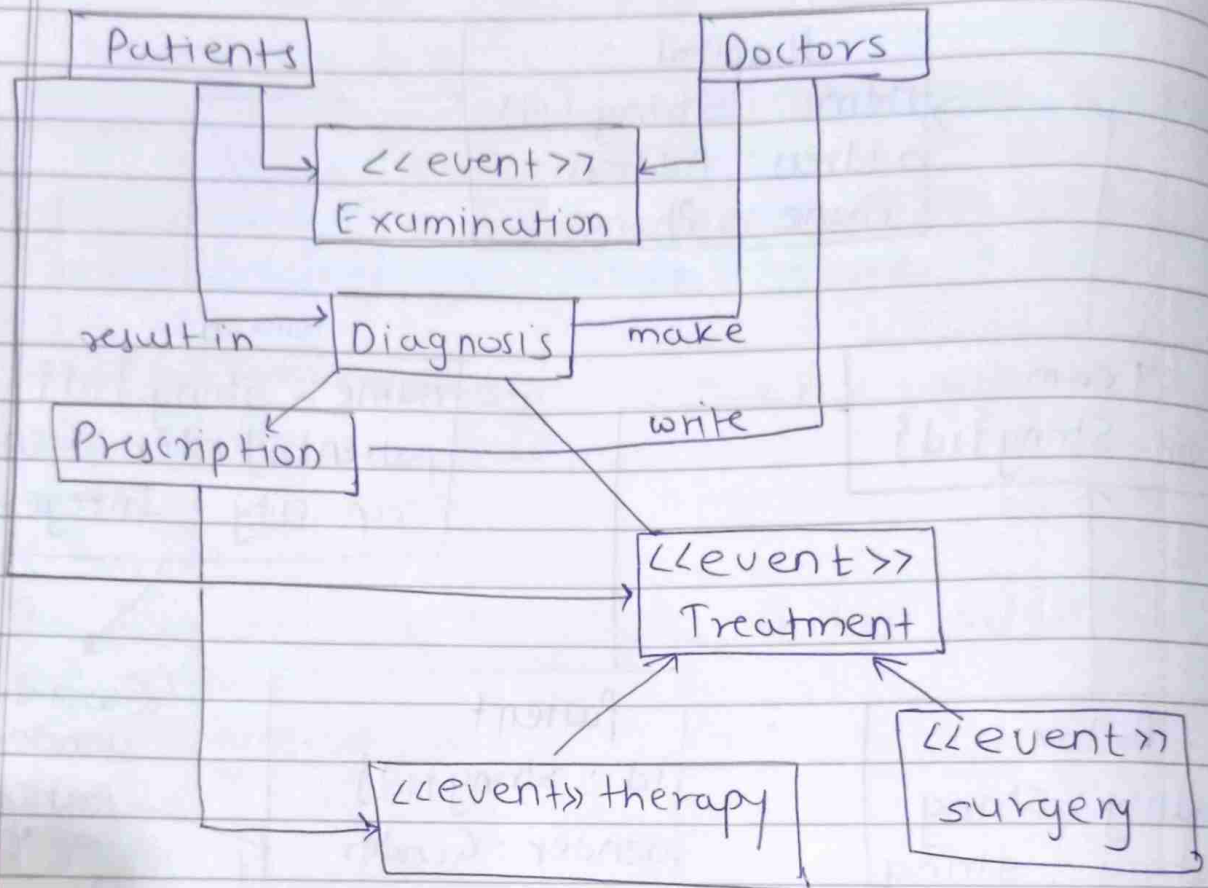
Class organizations.



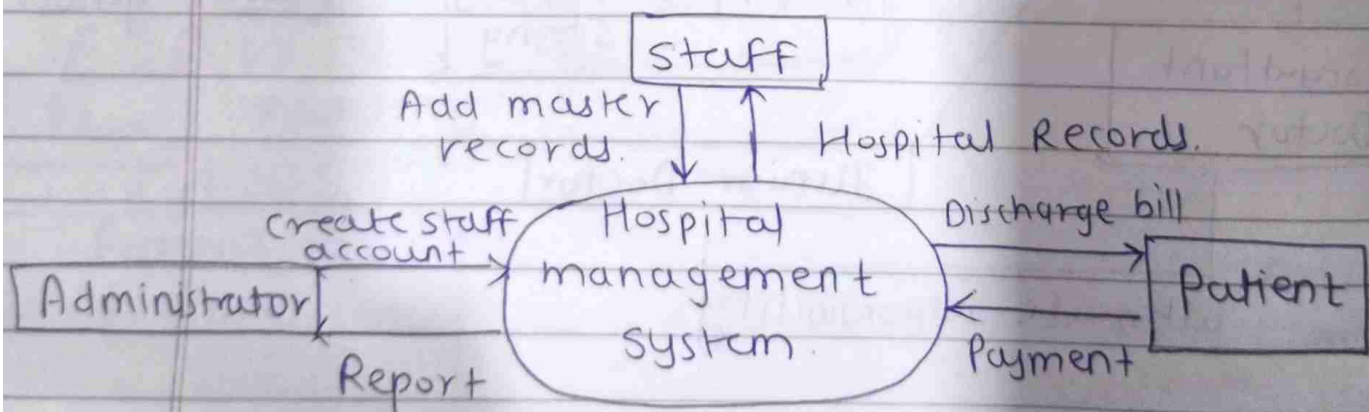
## Class words &amp; terms.



## Class treatments

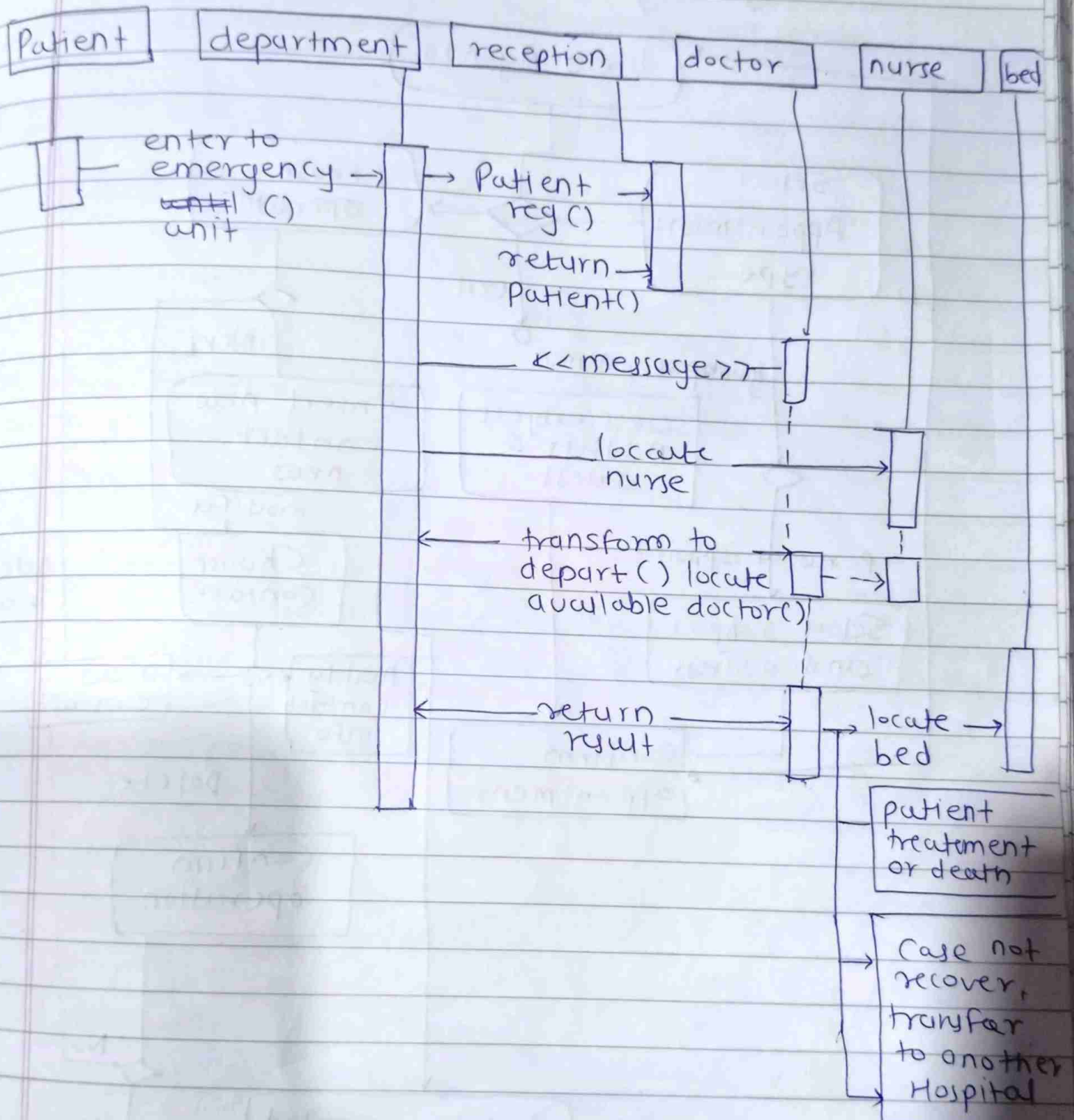


## Data Flow Diagram.





# Object Diagram.



# Activity Diagram.

