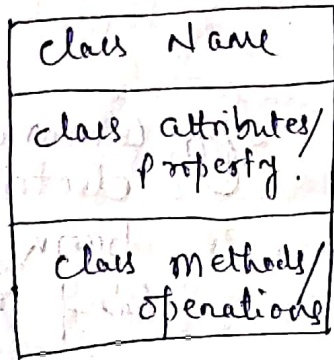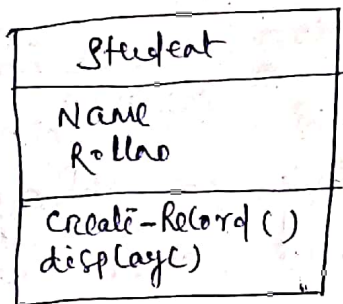# class Diagram ✓

*) CD comes under structural diagram

*) Structural diagram captures the static behaviour of the system.
   e.g. class diagram

*) object oriented view of a system can be represented by class diagram.

| class Name |
| --- |
| class attributes/ property. |
| class methods/ operations |

e.g.

| Student |
| --- |
| Name Rollno |
| Create-Record ( ) display( ) |

more generalization

| Student |
| --- |
| Name : String Rollno : int |
| Create-Record : int display ( ) : int |

Relationship
_____ ( Association)
————◇ ( Aggrigation) (HAS-A)
————◆ ( Composition) (IS-A)

more generalizn

| Student |
| --- |
| + Name : String − Rollno : int |
| # Create-Record : int |
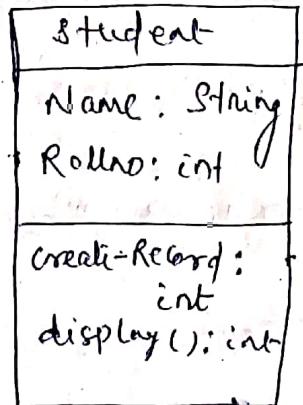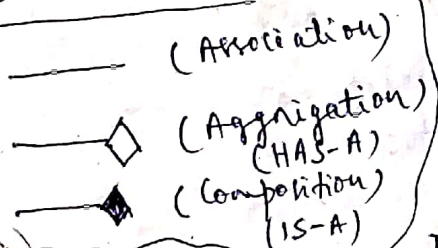
+ → Public

− → Private

# → Protected

~ → Package
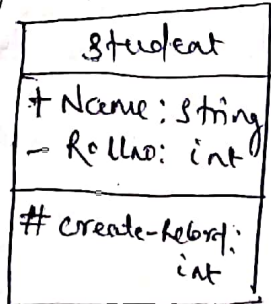
(final)

# Banking System

what are the classes

Bank
Branch
Account

**Account**
- accno : int
- balance : String

+ deposit()
+ withdraw()
+ addloan()
+ removeloan()

0..*     1

**Branch**
- branch-name : String
- code : number

+ AddAcc()
+ remAcc()

1

**Bank**
- name : String
- Location : String
- code : number

+ addBranch()
+ remove Branch()

**Loan**
- loanno : int
- amount : float
- type : string

+ addcustomer()
+ get Emi()

0..*

**Saving-Acc**
- minbalance : float
- opening-date : date

+ add customer()
+ remove customer()

0..1

1

**Current-Acc**
- minbal : float
- opening date : date

+ add customer()
+ remove customer

0..1

1

**Customer**
- customer ID
- name
- Address

1

1

# Library Management System.

**Book**
name: a string
auther: String
Id: int
+ add to findbook()

**Library**
- Name : String
- phone : int

+add book()
+remove book()
+add journal()
+remove journal()

1          1···*          0··6        0··12

1                    1th···*

1

**journal**
- name: String
- auther: String
- id: number

+ find journal ()

0··*

1··*

**library member**
- name: String
- id : number

+issue book ()
+ return book ()

**Student**
- name: String
- id: number

**staff**
- name: String
- Empid: number

- issue journal ()
- Return journal ()

0··*
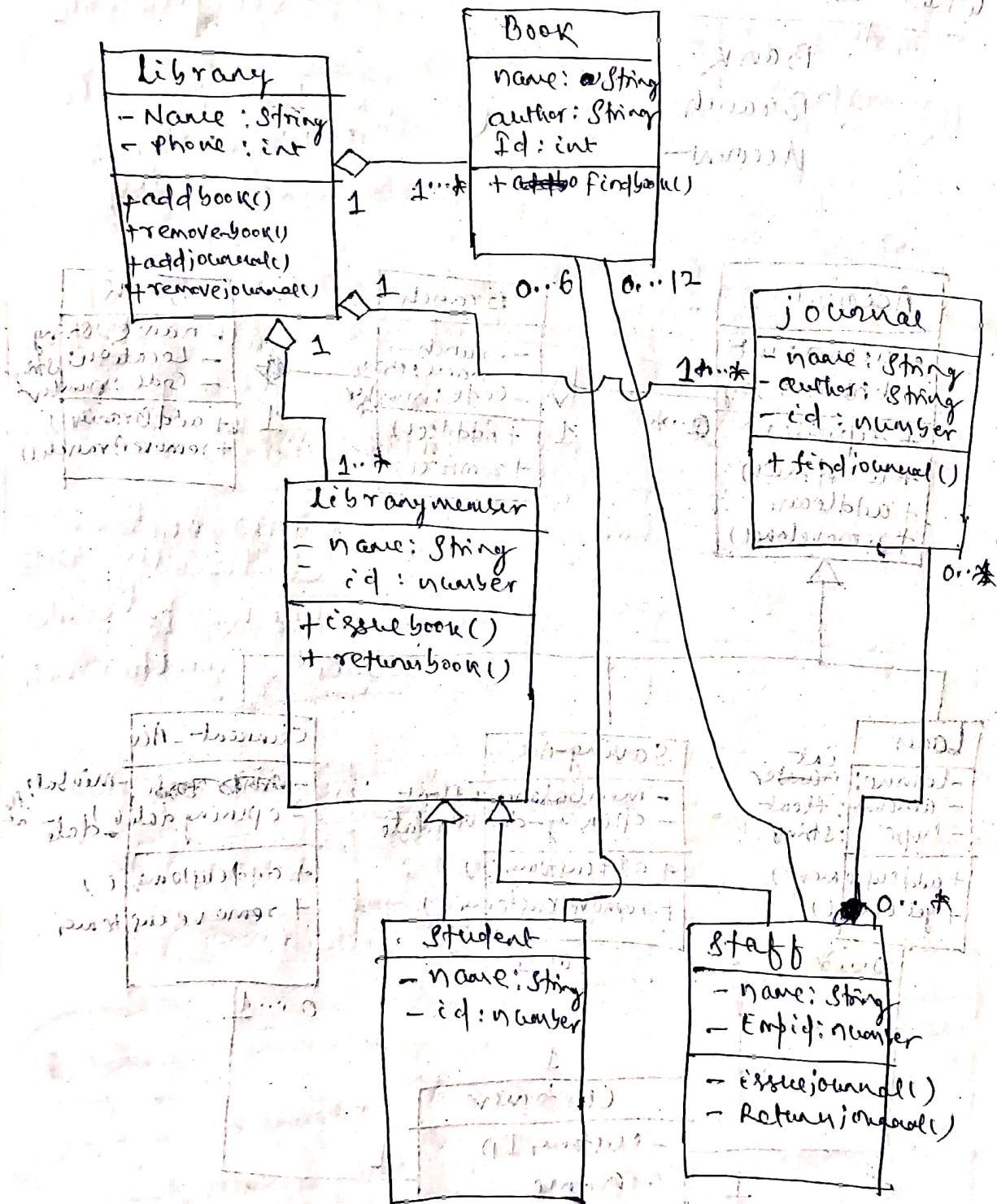
# Activity Diagram.
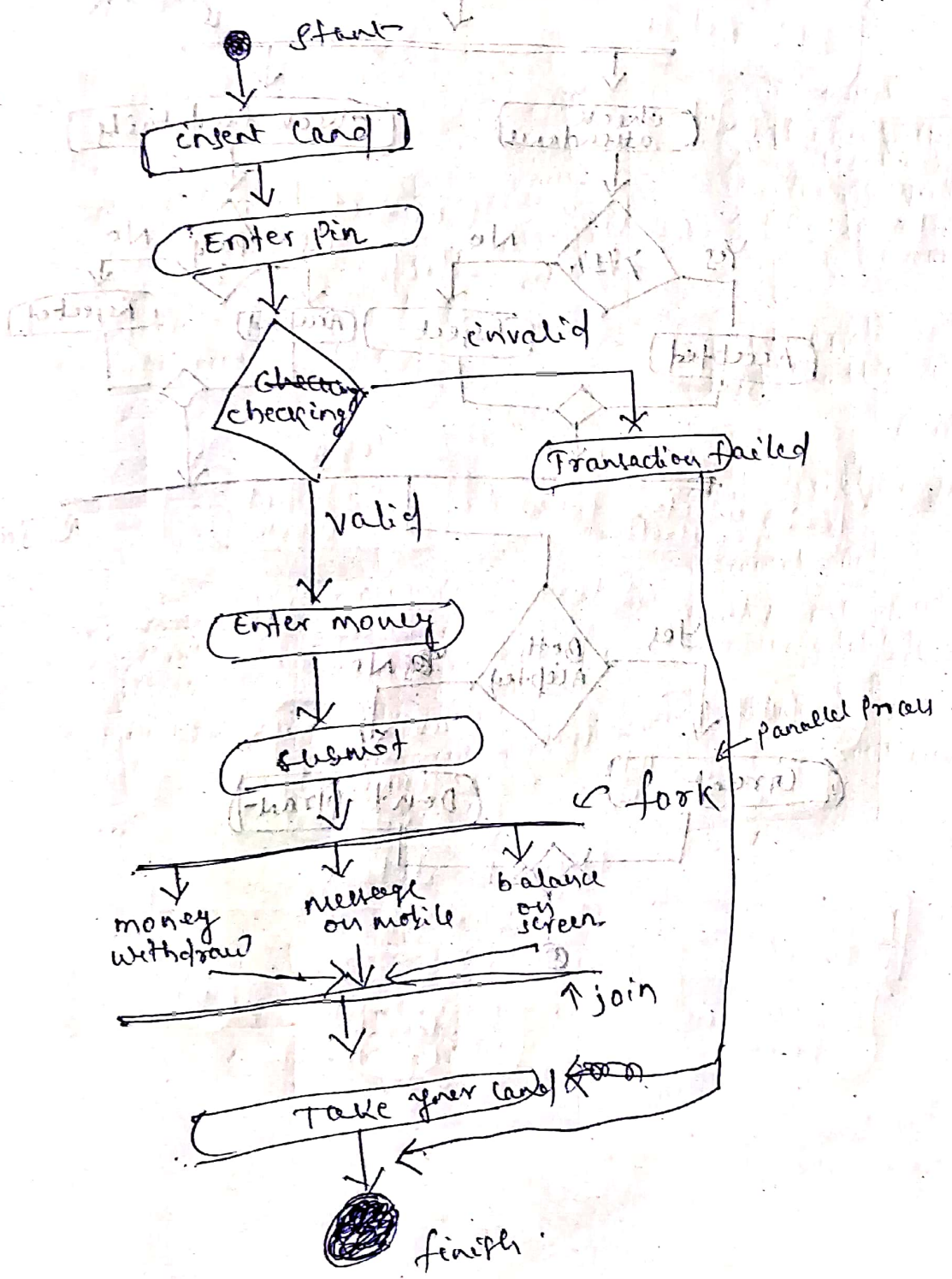
*) Comes under behavioural diagram of UML.

e.g.  ATM  Money  ~~withdrawal~~ withdrawal system.

start → Insert Card → Enter Pin → Checking

Checking → invalid → Transaction failed

Checking → valid → Enter money → submit → fork

fork → money withdrawl

fork → message on mobile

fork → balance on screen

join ↑

→ Take your card → finish

parallel process

fork

join

finish

# Student Scholarship



Submit student details

fork

check attendance

check feedetails

>75% — Yes / No

Paid? — Yes / No

Accepted

Reject

Accepted

Rejected

join

Both Accepted — Yes / No

Grant

Don't Grant

# Sequence diagram.

## Sequence diagram for Restaurant.

**customer** — **waiter** — **Kitchen**

- Menu
- serve menu
- order
- order food
- Activation time
- serve water
- pickup order
- serve order
- pay

## login pages System

**user** — **login page** — **Authentic**

- username
- password
- submit
- user login inform
- validate info
- accepted
- Redirect to target page
- Rejected
- invalid credentials
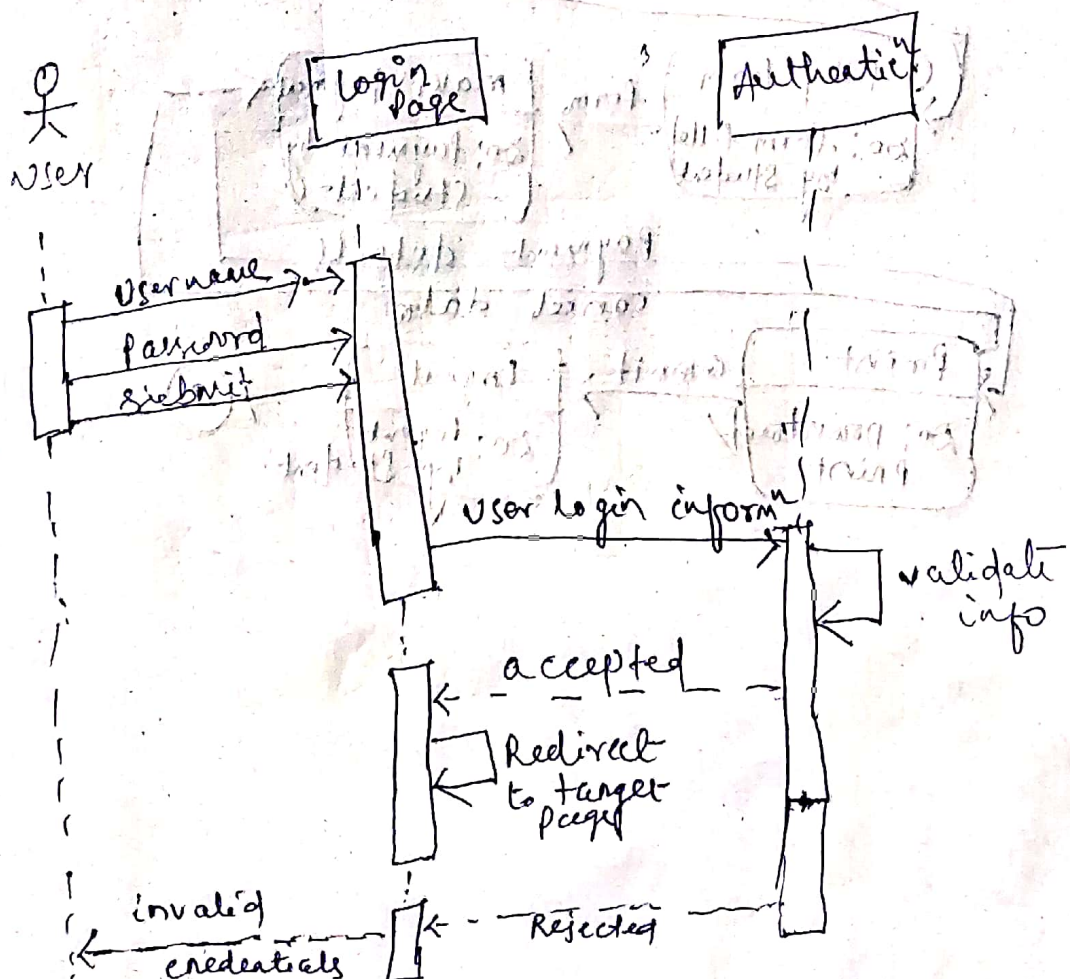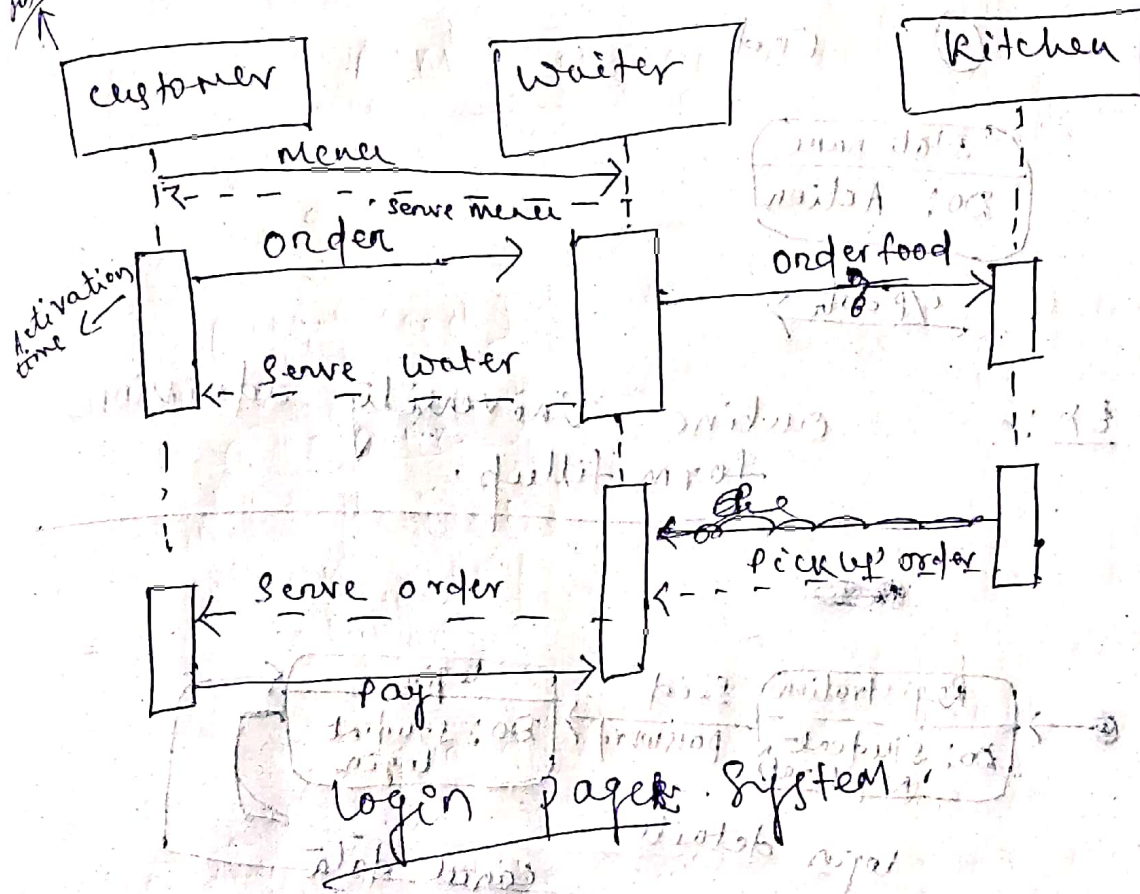
# State chart diagram

- ●    Start
- ◉    End

```
┌─────────────┐
│ State name  │
├─────────────┤
│ DO: Action  │
└─────────────┘
```
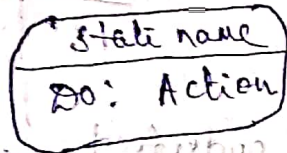
i/p data ──────►

Ep :-     online University admission form fillup.

```
           ┌────────────────────┐   S_id         ┌──────────────┐
  ●───────►│ Registration       │  password ────►│ login        │◄──
           │ DO: Student        │                │ DO: student  │
           │     Registration   │                │     login    │
           └────────────────────┘                └──────────────┘
                                                           │
              login details                      cancel data
```

```
           ┌────────────────┐   form    ┌──────────────────┐
           │ fillup form    │──────────►│ make payment     │◄──
           │ do: form filled│           │ do: payment by   │
           │     by student │           │     student      │
           └────────────────┘           └──────────────────┘

                    Payment details
                    cancel data
```

```
   ┌──────────────┐   commit   ┌──────────────┐
   │ Print        │───────────►│ logout       │────────►◉
   │ DO: Download/│            │ DO: logout   │
   │     Print    │            │    by Student│
   └──────────────┘            └──────────────┘
```