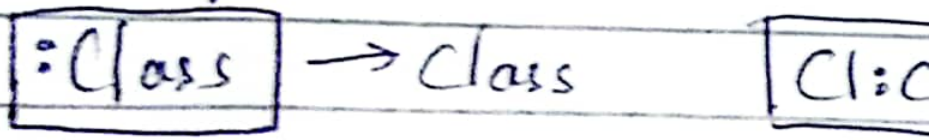
Covid-19 Survey

Sequence Diagram:- (UML)

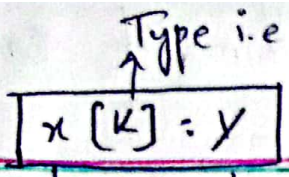
↳ Sequence of Events in your program.

↳ Represent message passing among different classes.

→ Dynamic



Objects can communicate with class.



<<interface>>

The complete abstraction is interface

How class communicate with another class?

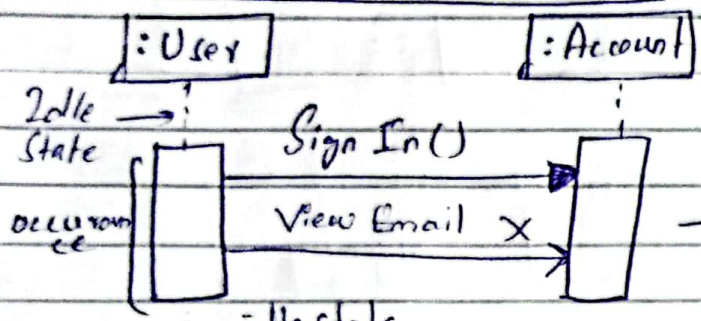
Messages :-

Synchronous ——▶

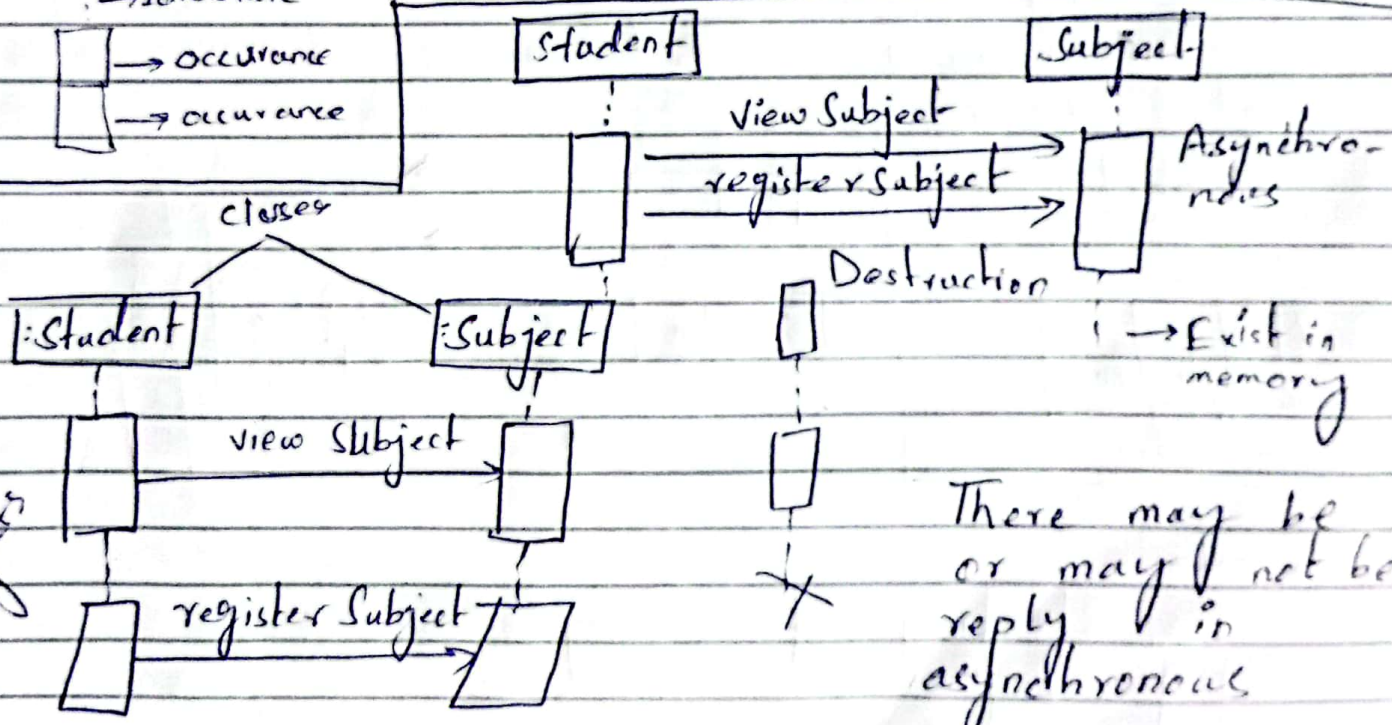
Asynchronous ——→

reply <- - - - - -

In a condition where a class can not ~~receive~~ send a reply until there is a reply from same class is known as synchronous.



→ life time of class/object



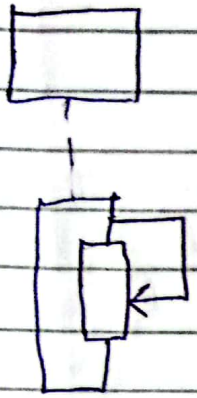
There may be or may not be reply in asynchronous

There must be reply in synchronous

A class call within it cause virus.

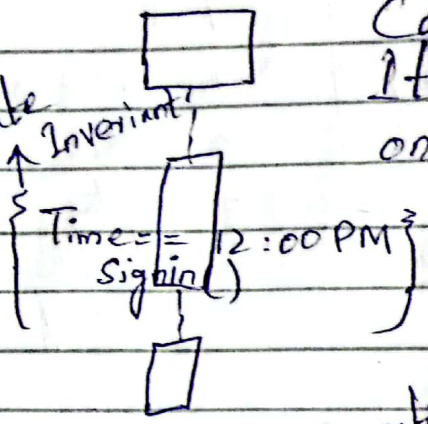
Recursion:-

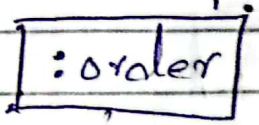
A class calls itself or a class calls another class, and that class calls this one.



State Inveriant:-

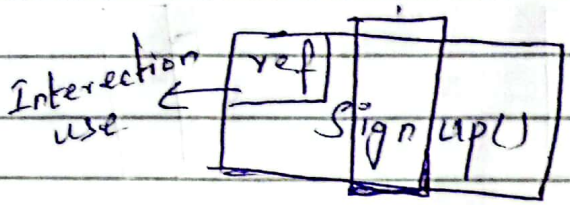
Condition on lifetime.
It must be done on lifetime of system.

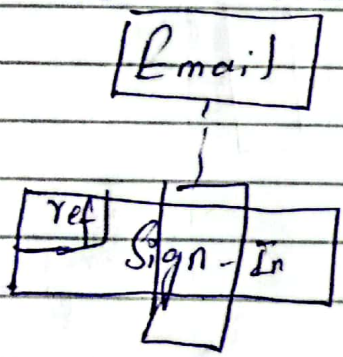


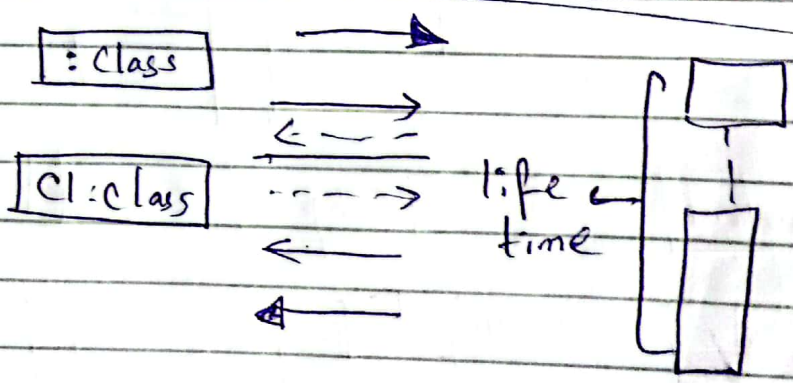


Interection Use:- Condition

Untill first event is not occureab you can not move to next event.

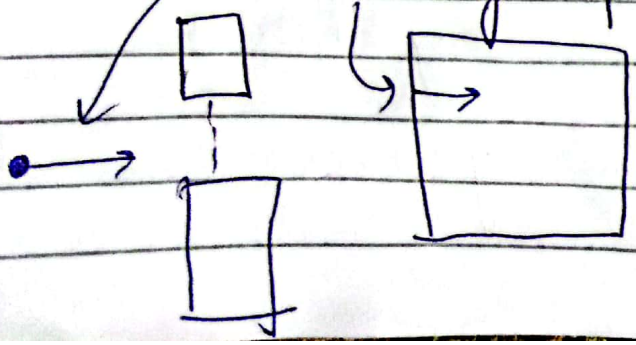




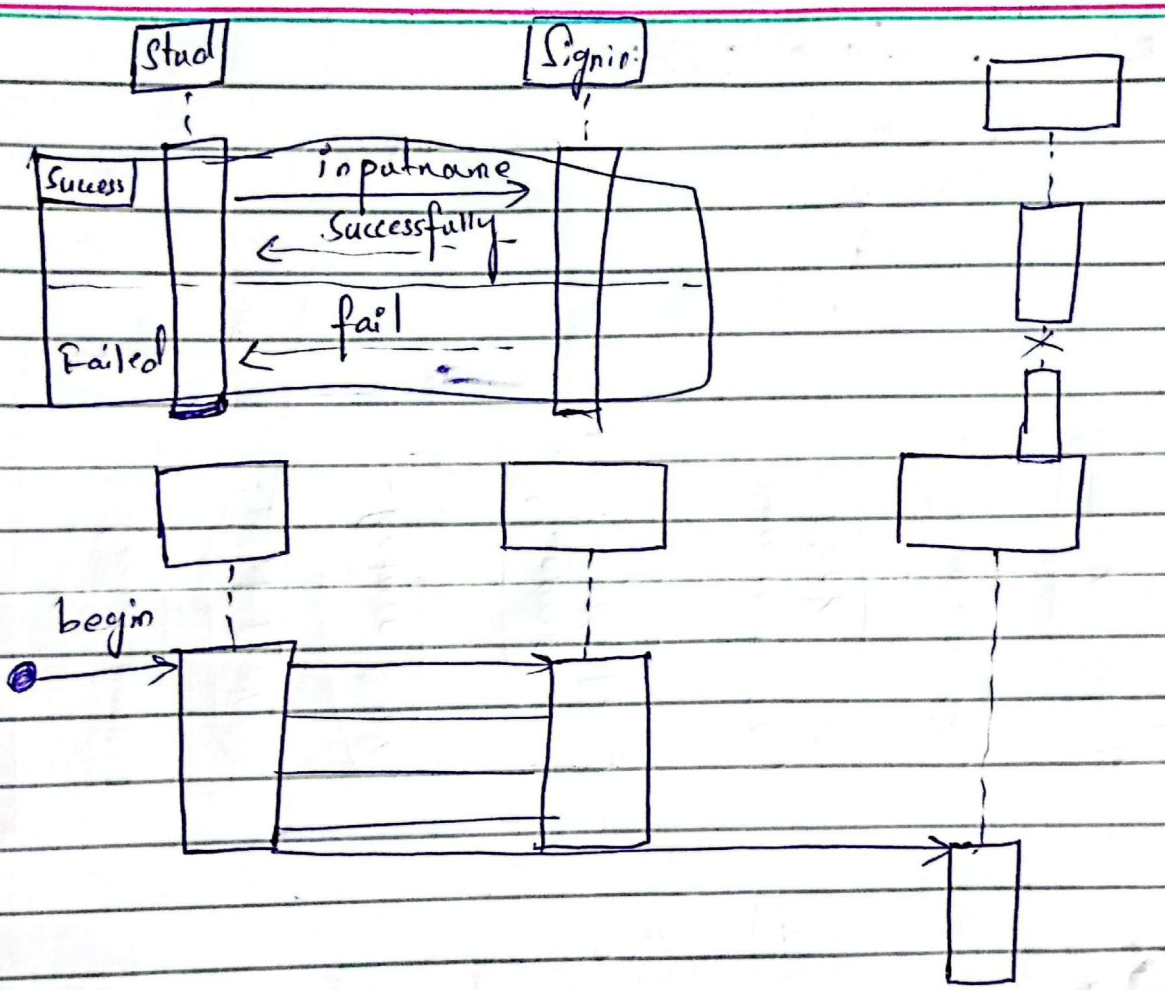


Gate:-

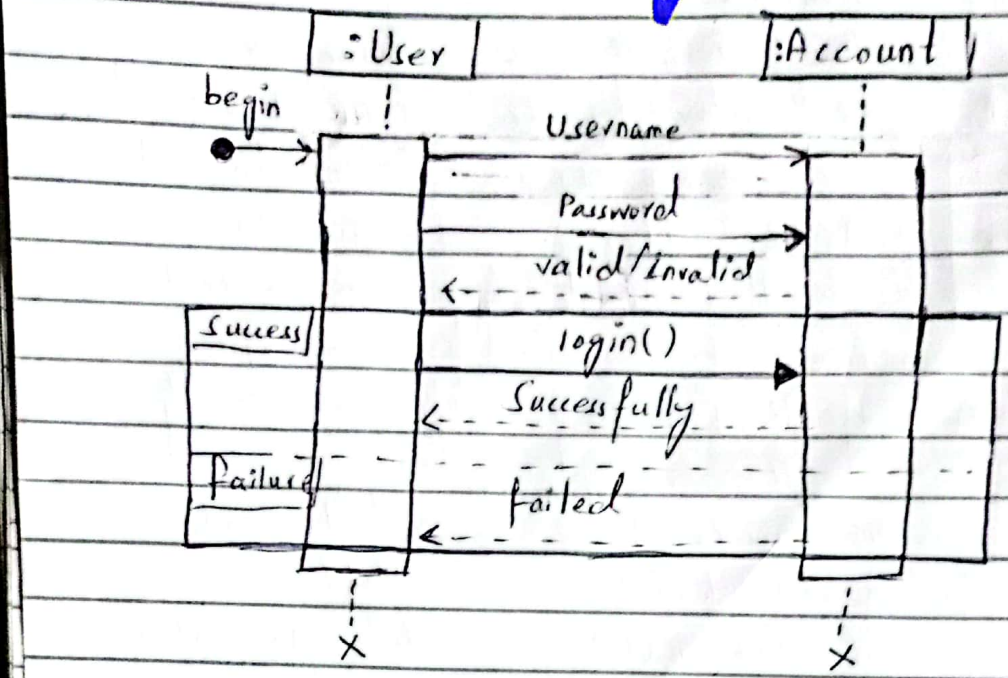
The starting of Diagram.

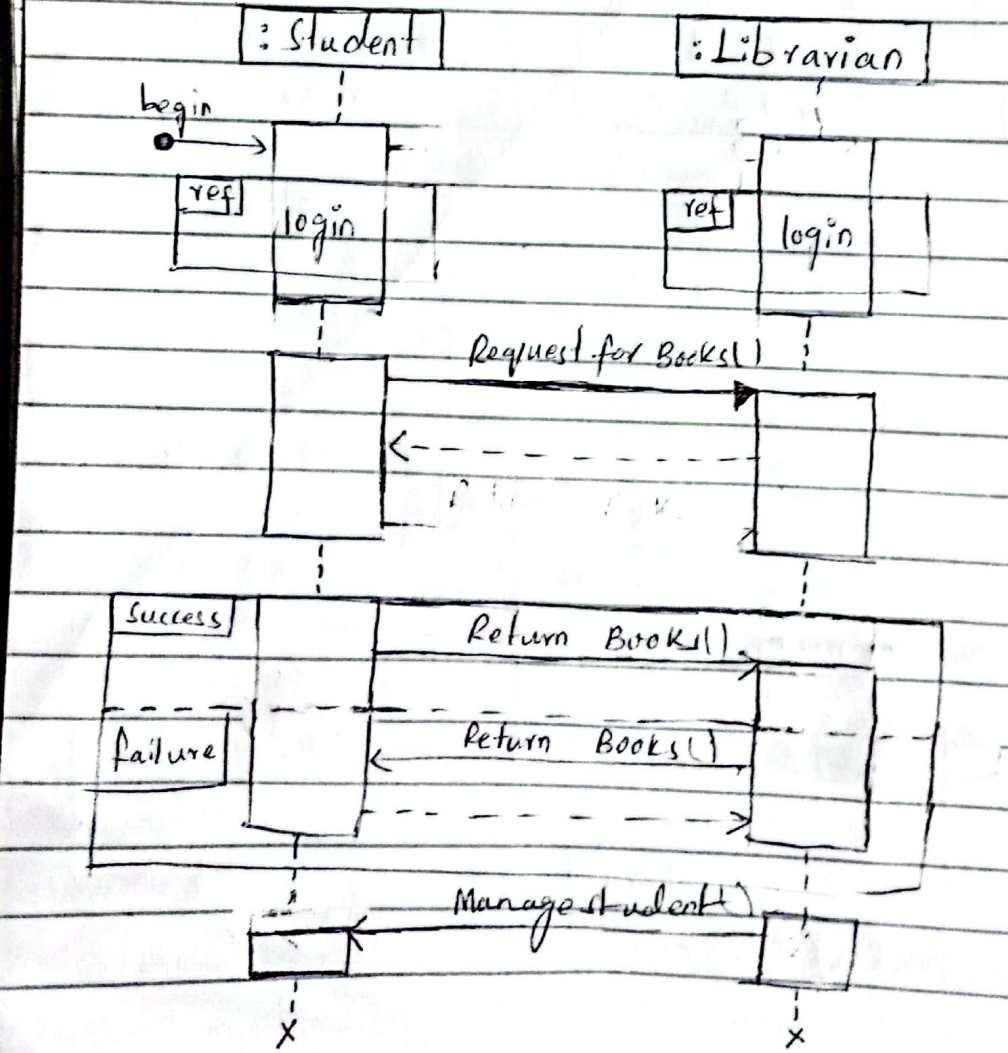


if - else



Sequence Diagram :-



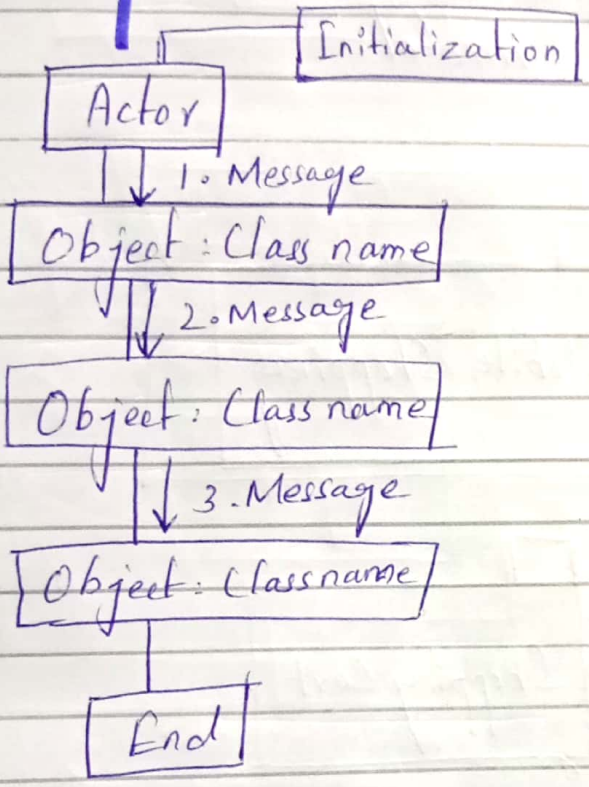


Collaboration Diagram:-

↳ Relationship and interactions among software objects

↳ Dynamical behavior of a particular use case.

Components:-



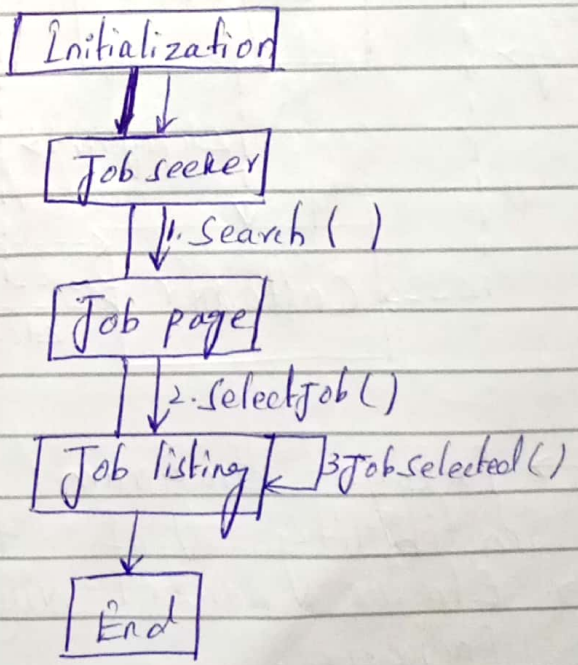
→ Collaboration diagrams have no concept of the asynchronous messages, since its focus is not on the messages ordering

→ Collaboration diagram is vertically representenal.

→ There is messages numbering

→ In rectangular boxes member functions will not be written.

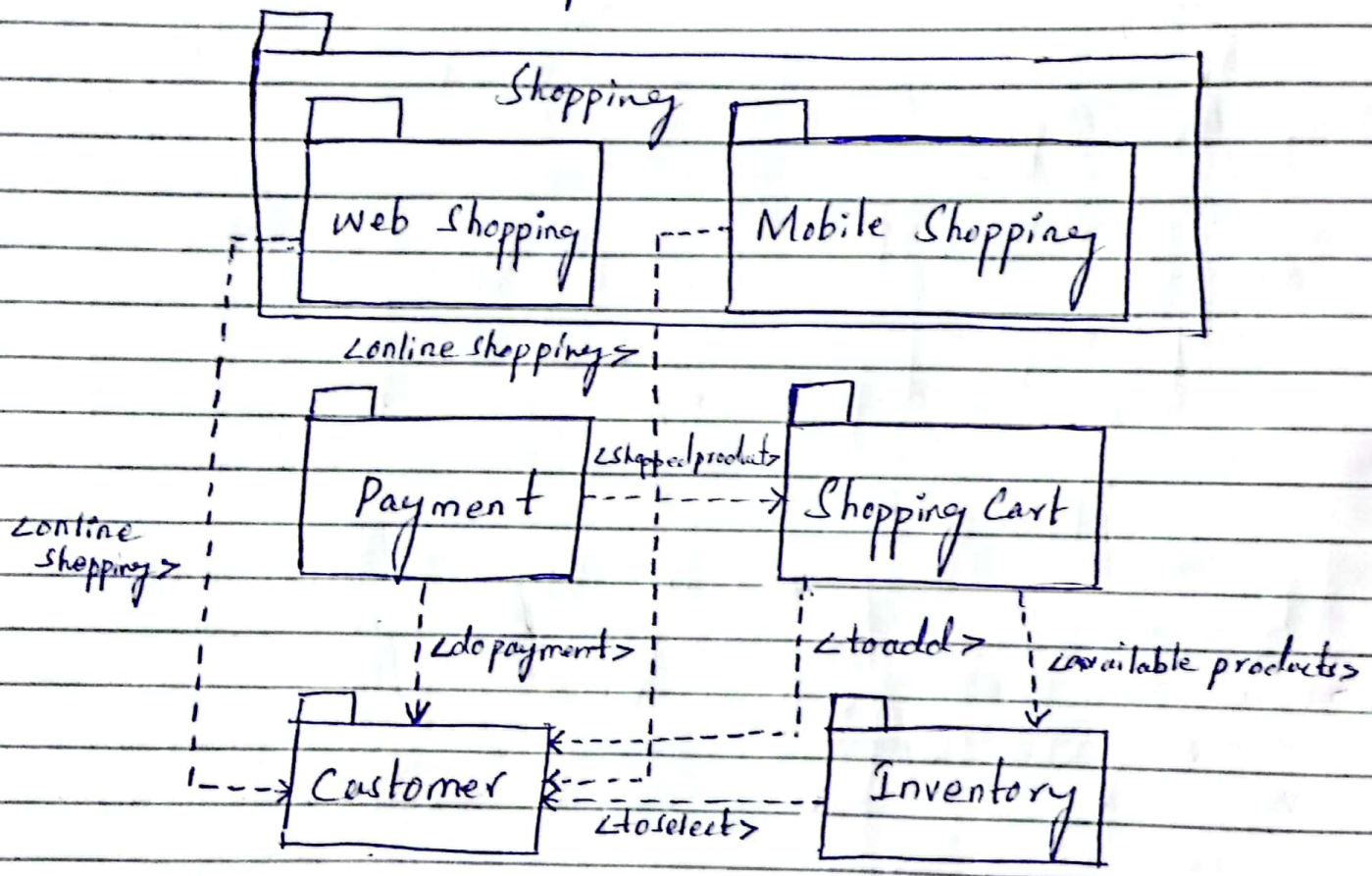
→ There is enating and starting in collaboration diagram.



Package Diagram:-

→ Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of Packages.

→ It depicts the dependencies between Packages that make up a model.



→ Package diagram is created to increase readibility of the program

→ Classes names will be written in Packages.

Deployment Diagram :-

→ How can we deploy a system in real

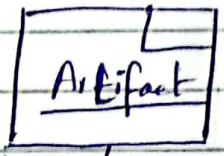
user → modem → Internet → website → Servers → Database

How we deploy system after its completion.

① Artifact
(Real world entity, Source file, exe file, DB table, output file, DLL etc)

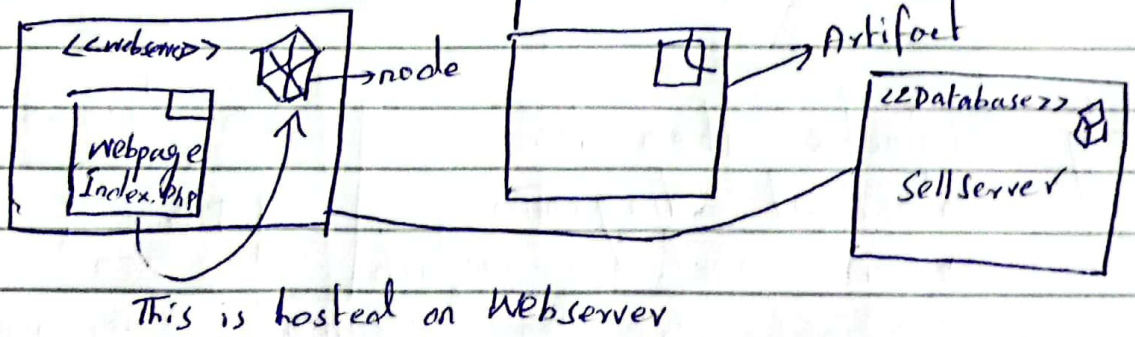
for games

② Artifact Instance.

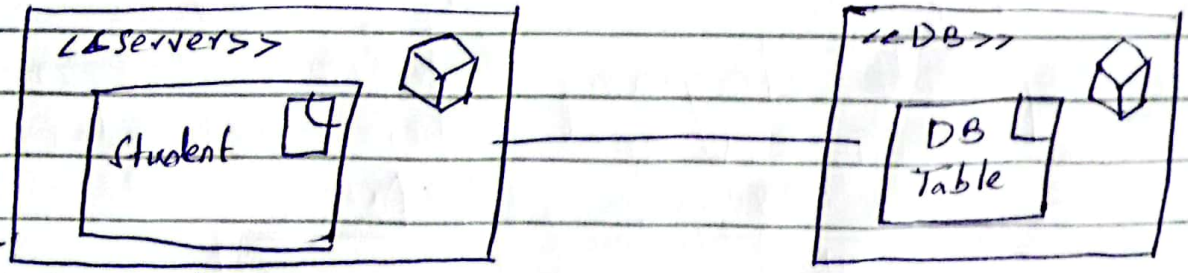


③ Node : On which Artifact execute

A machine on which things executed

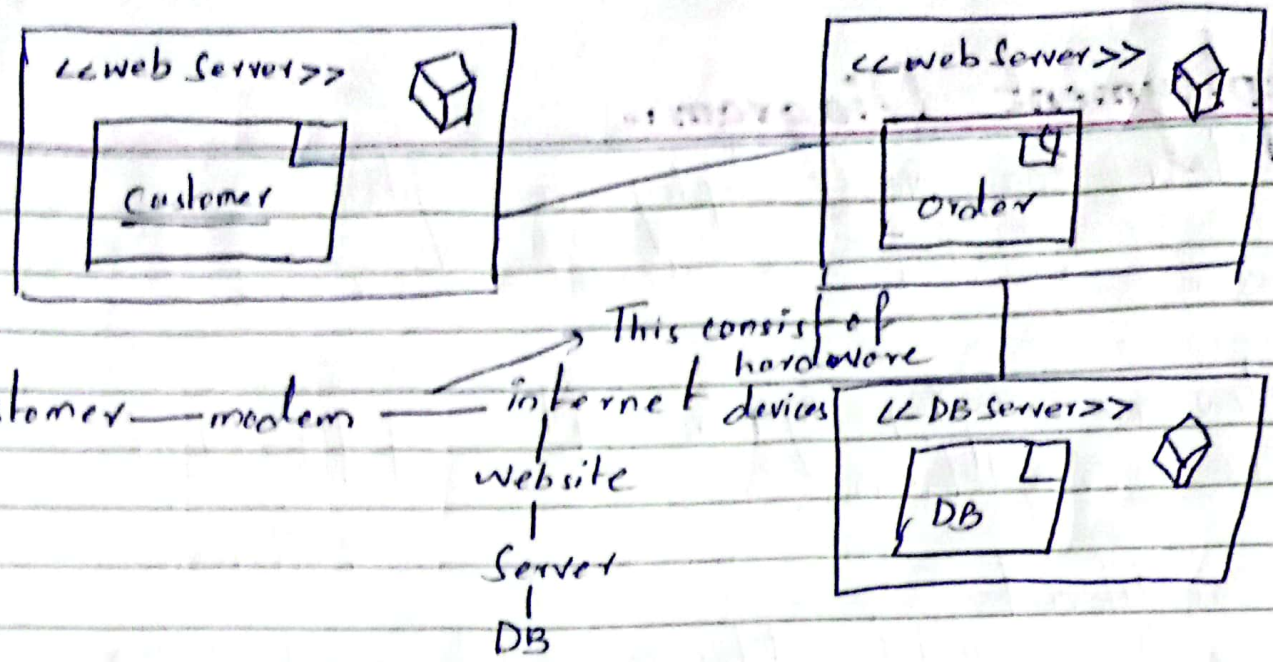


This is hosted on Webserver

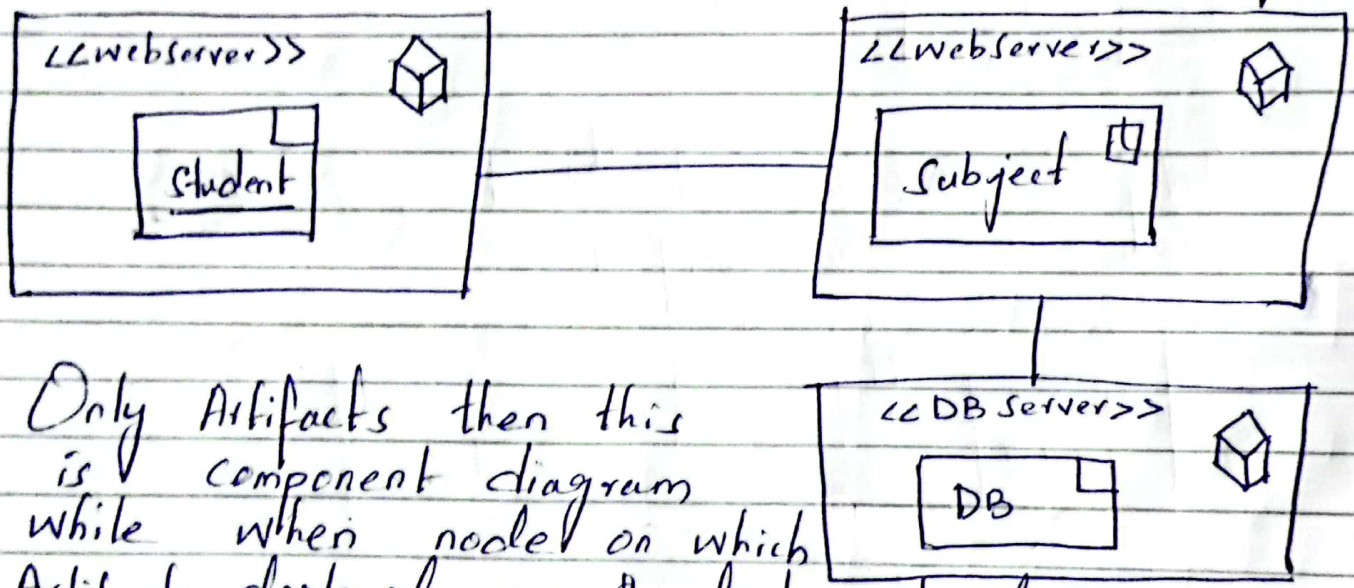


node = server

node = DB

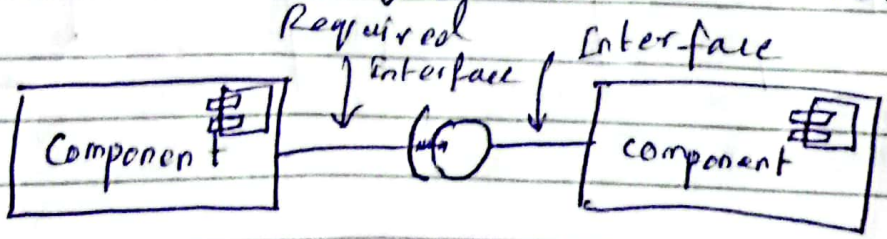


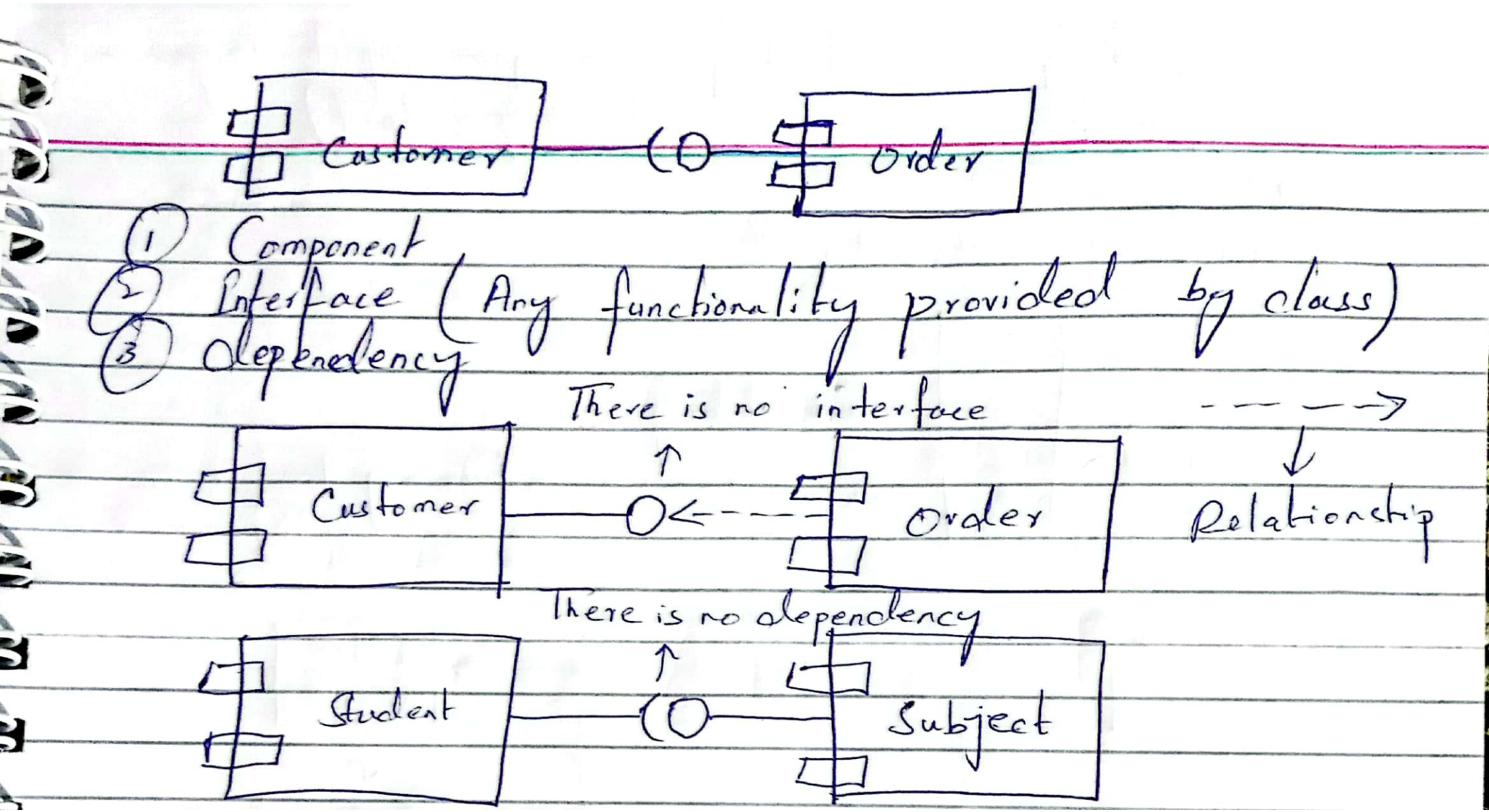
(*) Multiple Artifacts can not be on a single node.
(*) On a node there can be multiple Artifacts.



Only Artifacts then this is component diagram while when nodes on which Artifacts deployed are deployment diagram.

In component diagram, hardware components are shown and their relationship.



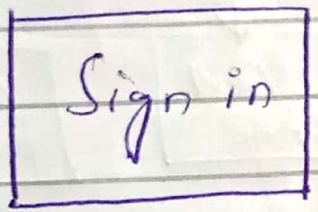


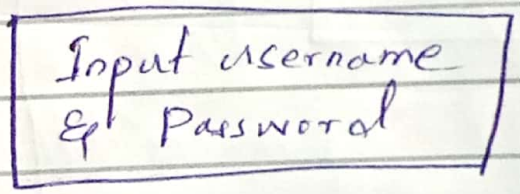
Activity Diagram:-

→ Dynamic diagram

Class
Sequence
Activity
→ Most important Diagrams

→ How activities flows in system

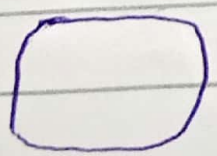




System = State
User = Activity

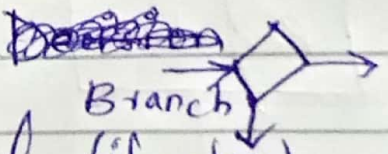
Subject is being registered → State
Subject is registered by Student → Activity

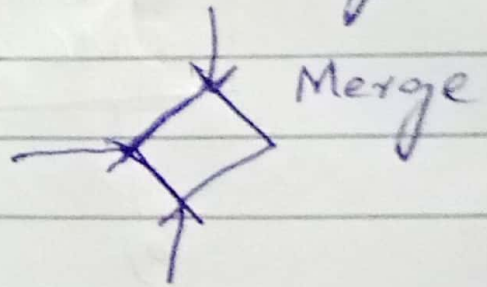
Start Activity

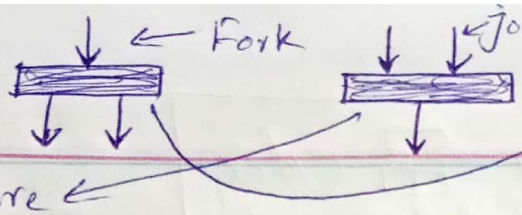


End

Flow →

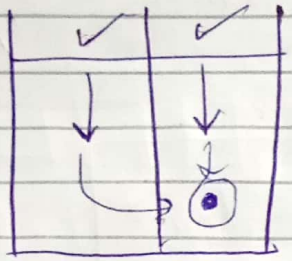




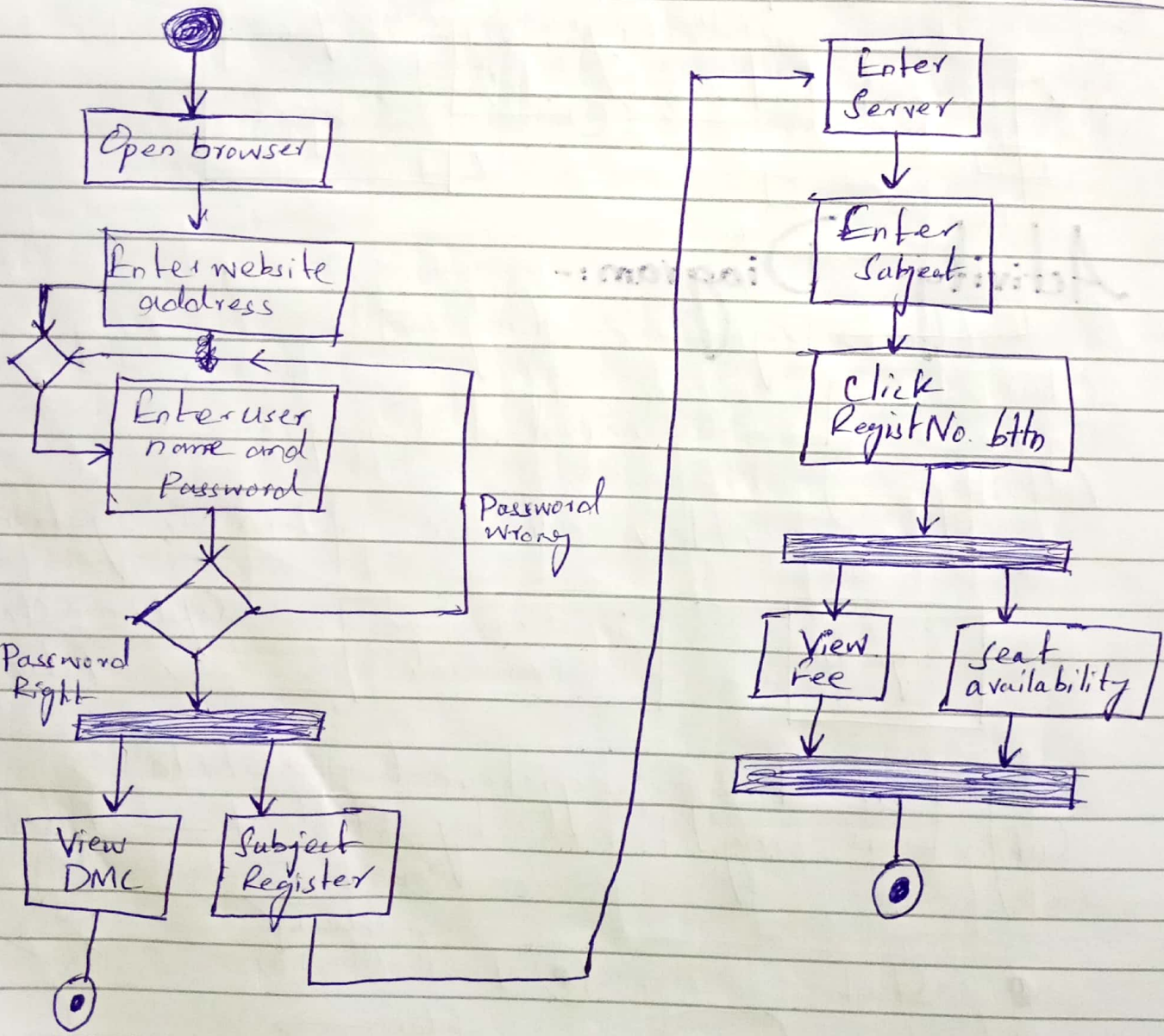


(To give input to generate many output)

Fork are combined in one output

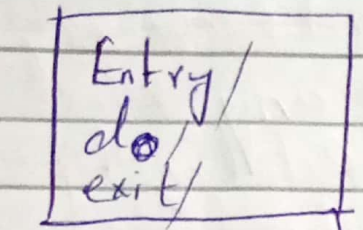


Swim lanes (Load may be in single or on both)

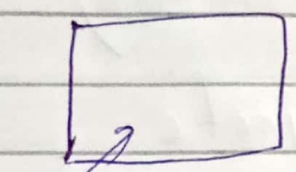


drawio

After each activity state of the system changes



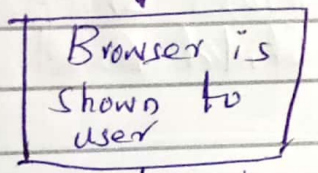
Entry/ Login page shown
do/ Username and password is entered
Exit/ Credential are forwarod
to DB



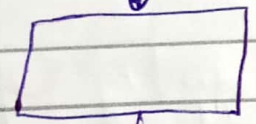
do

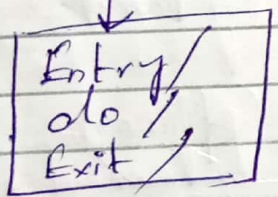
Detailed

User open browser



Enter Credential





Testing :-

* The fault done by developer is called error.
# The error done by human in software of developer is fault.
Presence of error in software is known as bug.
Deviation of software from expected result, is known as failure.

---

* Error (Error is human action that produces incorrect result)
# Fault (State of Software caused by error)

Testing life cycle :-

Project Initialization
↓
System study
↓
Test plan
↓
Test cases Design
↓
Test cases execution
↓
Report Defects
↓
Regresion Testing
↓
Analysis
↑
Report (A document contain all the details)

Levels of Testing :-

1. Unit Testing
2. Integration Testing
3. System Testing

White Box → Internal + External Testing (DB Testing)
Black Box → (External Testing) → Front end
Gray box (First External then Internal)

Integration Testing :- Top-down, Bottom up

Top-down
System
↓
Module
↓
Sub_mod
↓
Unity
↓
Stubs and Drivers
↓
Bottom up (Dummy unit)
Top-down (Dummy modale)

Bottom-up
Unity
↓
Sub-mod
↓
Module
↓
System

Stubs (dummy unit module)
To test the units you create dummy module

Types of Testing :-

1) Unit Testing
2) Integeration
3) System
4) Usability (Must be easy to use i.e GUI) (look in feel, Speeded)
5) Performance (loading, Stress, Data) (loading = lower, Stress = upper)
6) Security (Unauthorized access, Virus, Username, Password)
7) Smoke (Test the system not to break the system)
8) Alpha (Test on developers ride, controlled conditions)
9) Beta (Outside the Software house, Uncontrolled)
10) Acceptance
11) Regrenion (Test a function if bug identified, correct it, test again)
12) Monkey

In Performance you break the system.
Stress is from upper limit.
load is from lower limit.

Smoke:-
Small inpat, Test main functions only

Alpha:- (Employees of Software house)
It is performed by the persons not the developer, but work at software house.

Beta :-
These are general public not the people of Software House. They are not much expert. They may or may not be expert.

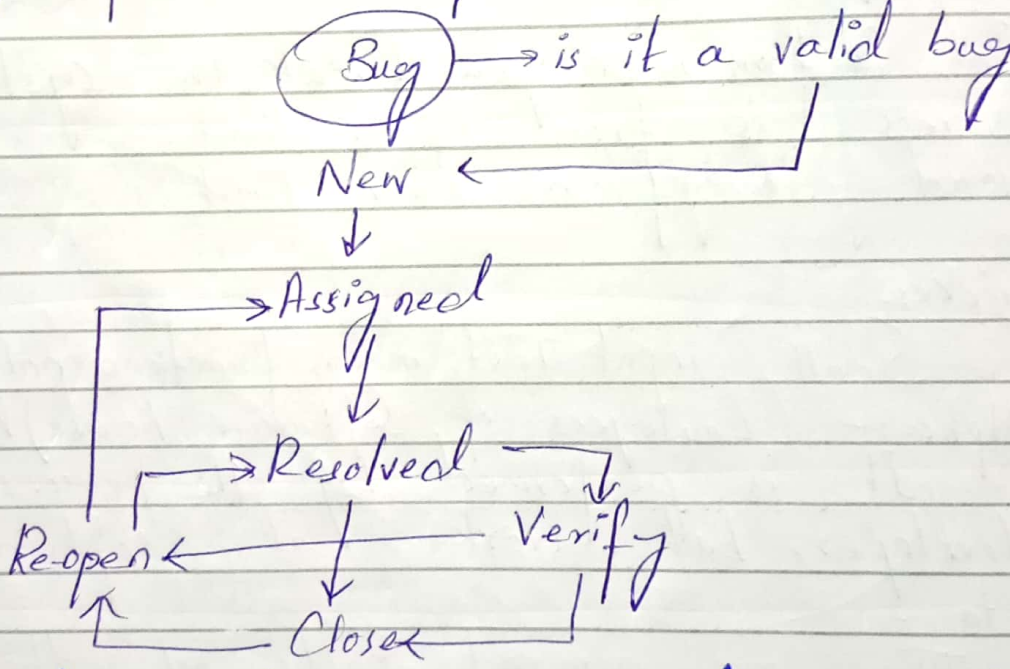
Acceptance Testing:-
Testing of all user level requirement.

Regrenion :-
To do testing from 1st step if you found a small bug.

Monkey :-
Test the software randomly, Testing is on small modules.

Verification :-
Verification is the product right. (Specification, Guidelines, inspection etc.)

Validation :-
Validation is the right product (Meeting of User requirement)



Functional Testing :-
- Initialization
- Usability
- Smoker
- Regrenion

Non-Functional Testing
- Performance
- Stress
- Security
- Accenibility

- Monkey
- Destructive
- Recovery
- Acceptance
- Load Testing
- Data Testing

In destructive testing you loading to much to destroy or crash the system and in Recovery you recover your destroyed or destructed system.

Unit Level :-

Email [ ]
Password [ ]

→ ~~Password~~ Email shall be correct

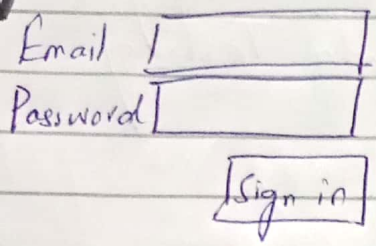
→ Password shall have Exact 8 characters
- Alphabets
- Numbers
- Special characters

It Should be in Separate table.

Test Data	Expected Result	Actual Result	Result
abc@gmail.com	Accepted	Accepted	Pass
abc@	Rejected	Rejected	Pass
abc@gmail	Rejected	Rejected	Pass
abc@.com	Rejected	Rejected	Pass
123 ABC*+	Accepted	Accepted	Pass
123 ABC	Rejected	Rejected	Pass
ABC	Rejected	Rejected	Pass
123	Rejected	Rejected	Pass
12345678	Rejected	Rejected	Pass

Test Data	Expected Result	Actual Result	Result
User Click on Sign in button	User Should be Signed in	Accepted	Pass

Integration Testing :-



Test Data	Expected	Actual	Result
User Sign-Into the system (after inputing Credential)	User Shall Sign-In	User has Signed in	Pass
User Account does not exist	User shall move to sign up page (should not sign in)	Successful	Pass
User entered wrong Email	User should not Sign in.	Successful	✓
User entered wrong Password	User Should not Sign in.	"	✓
User forgot Email	User should not Sign in.	"	✓
User forgot Password	User shall move to forgot Password page (should not sign in)	"	✓
Email Field is empty	User should not Sign in.	"	✓
Password Field is empty	User should not Sign in.	"	✓

System level Testing:-

Test Case ID: 001 Test Case Description : Sign_In
Created by : Zeeshan Reviewed by : Ali: Version : 1.1

QA Test log:- Ali reviewed by testing performed by Zeeshan and ask him to revise!

Tester name : Zeeshan Date : 20/6/2022 Test : Pass

Pre-requisite :- i) internet Connection
ii) Account exists

Test Data i) Email= Zeeshan@gmail.com
ii) Password= 123@abc1

Test Scenario :- User must be able to sign into the system

Sr. #	Steps	Expected Result	Actual Result	Pass/Fail/Not executed/Suspended
1.	Enter Email is pass	Credential entered	As expected	Pass
2.	Click sign in button	logged-in	As expected	Pass

Post Condition : User is signed-In and able to search products.