# Unit 5 :- Software Quality Assurance

## What is Software Quality Assurance?

Software quality assurance ( SQA ) is the ongoing process that ensures the software product meets and complies with the organization’s established and standardized quality specifications. SQA is a set of activities that verifies that everyone involved with the project has correctly implemented all procedures and processes.

* It is an umbrella activity which focuses on improving the process of development of software so that problems can be prevented before they become a major issue

## Major Software Quality Assurance Activities

* Management plan
* Check points
* Multi testing strategy
* Measure change impact
* Manage good relationship between programmers and QA
* Formal Technical Reviews
* Software Engineering Techniques
* Manage whole projects documentation

## SQA Techniques

* SQA Techniques - QA professionals inspecting the work
* Reviewing - stakeholders meet to examine the product
* Code Inspection
* Design Inspection
* Simulation
* Functional Testing
* Stress Testing
* Walkthroughs – problem solving
* Six Sigma – means 99.76 % defects free product

## What is Cost of Quality?

* Cost of quality (COQ) is defined as a methodology that allows an organization to determine the extent to which its resources are used for activities that prevent poor quality, that appraise the quality of the organization's products or services, and that result from internal and external failures.
* **There are four categories to measure cost of quality:**
  + Prevention costs – cost for experienced developers
  + Detection costs – cost of find bugs and tester and test cases
  + Internal failure costs – cost of bug fixing before delivery
  + External failure costs – cost of defect fixing after deliver and also include support

## Quality Standards

-iso

-cmm

-six sigma

## ISO 9000 Standard

Iso – international organization for standardization

* Verify quality of product during development

- 9001 iso use for QMS (quality management system)

- ISO 9001 : engaged in design, development, production, and servicing

- ISO 9002 : not involved in design but need in production

-ISO 9003 – involved only in the installation and testing

## Process of ISO 9000 certification

•Application: Once an organization decided to go for ISO certification, it applies to the registrar for registration.

•Pre-Assessment: During this stage, the registrar makes a rough assessment of the organization.

•Document review and Adequacy of Audit: During this stage, the registrar reviews the document submitted by the organization and suggest an improvement.

•Compliance Audit: During this stage, the registrar checks whether the organization has compiled the suggestion made by it during the review or not.

•Registration: The Registrar awards the ISO certification after the successful completion of all the phases.

•Continued Inspection: The registrar continues to monitor the organization time by time.

## Read CMMM model from the PPT.

* Imp for reading please note it

# Unit 1 :- Project Management

## What is project?

* A project is a group of tasks that needs to be completed to reach a clear result and also define as a set of input and output for reach goal.

## Software project management

* Software project management is an art and discipline of planning and supervising software projects.

## Prerequisites of Software Project Management?

* Time
* Quality
* Cost
* These three factors are connected with each others.

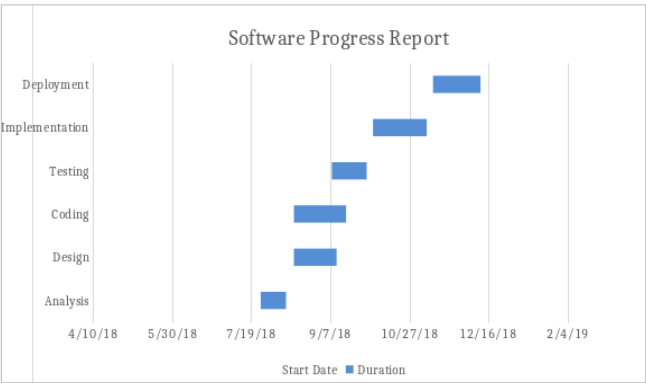
## Project Manager

* A project manager is a character who has the overall responsibility for the planning, design, execution, monitoring, controlling and closure of a project.
* Main role of project manager is handling entire things and also manage whole project and also communication between client and team.

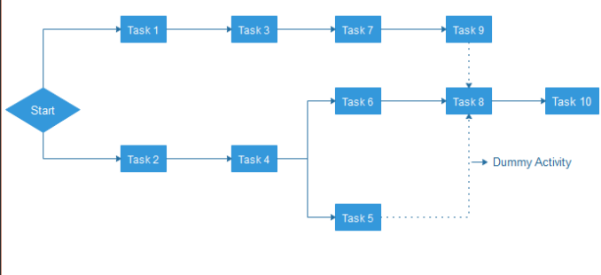
## **Activities of Project Management**

* Project tracking and control
* Scheduling management
* Scope management
* Software configuration Management
* Resource management
* Communication management
* Risk management
* Estimation management

## Project Management Tools

1. **Gantt Chart**

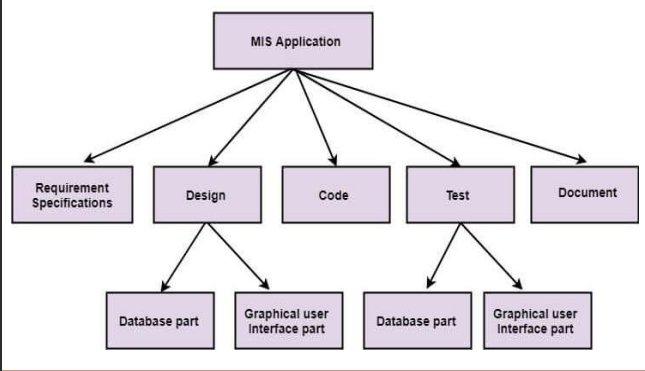
* showing activities displayed against time.
* It is mainly used for manage activity against time !!
* Activities represented by bar.
* Developed by Henry Gantt
* Also known as project scheduling chart

1. **PERT Chart**

* acronym of Programme Evaluation Review

Technique.(short technique for review of task)

* U.S. Navy to handle the Polaris submarine missile programme
* PERT chart represented as a network diagram concerning the number of nodes, which represents events.

1. **WBS (Work breakdown structure)**

* Divide whole big application into small parts

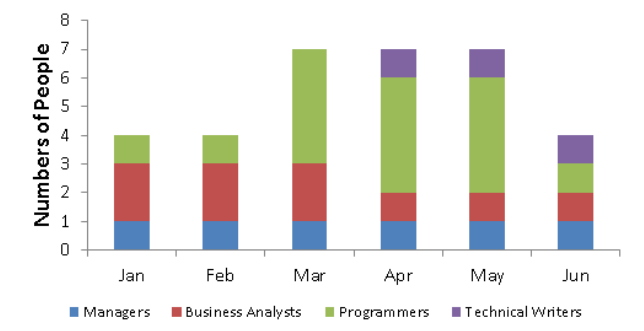
•**The top-down approach** – the WBS derived by crumbling

the overall project into subprojects or lower-level tasks.

**•The bottom-up approach** - is more alike to a brainstorming

exercise where team members are asked to make a list of lowlevel tasks which is required to complete the project.

1. **Resource Histogram**

****

In particular time how many   
resources used by team.

**5) Critical Path Technique**

**-** critical path means longest path for completing given task

## Measure

**Measure** : provides the **quantitative indication of the size of some attribute of a product** or process

Eg: Number of errors in a module.

**Measurement** : an act of determining measure.

## Metrics

**Metric** : a **quantitative measure of the degree** to which the system, component or process possesses a given attribute.

Eg: no. of errors per module

26 errors in module 1

32 errors in module 2

16 errors in module 3…..

* **Type of Metrics** 
  + Product Metrics - describe the characteristics of the product
  + Process Metrics – improve software development and maintenance
  + Project Metrics - describe the project characteristics

## Indicators

**Indicator** : It is a metric or combination of metrics that provide insight in a software process, project or a product itself.

## Software Measurement Principles (read from ppt also)

* Formulation - metrics appropriate for the representation of the software
* Collection - accumulate data required to derive the formulated metrics
* Analysis - computation of metrics
* Interpretation -
* Feedback

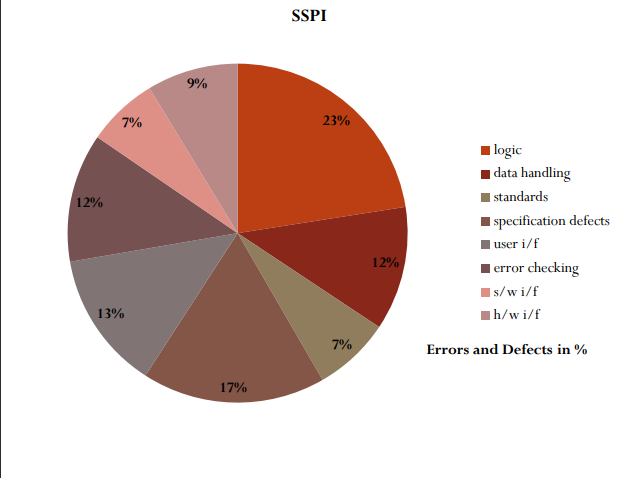
## Size Oriented of Metrics

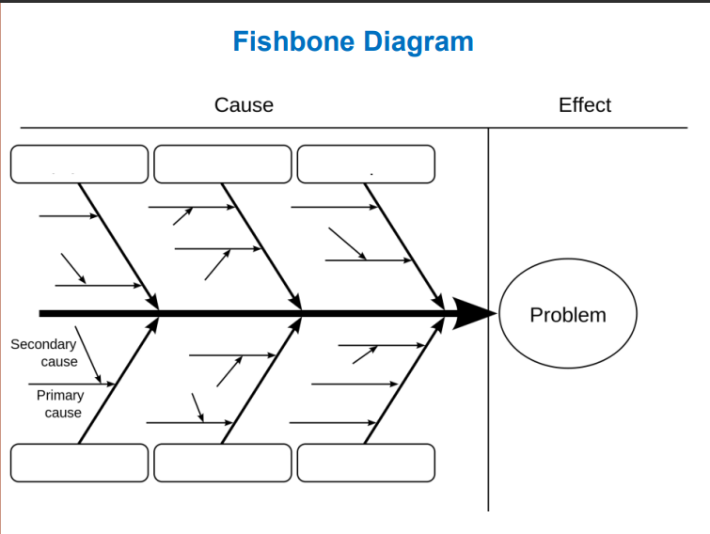
* measures by considering size of the software that has been produced
* built on past experiences of organizations.
* size measurement is based on lines of code computation.
* Based on the LOC/KLOC count of software

## Function Oriented Metrics

* Based on “functionality” delivered by the software as the normalization value
* Function point
* independent of the programming language

## **SSPI – Statistical Software Process Improvement**



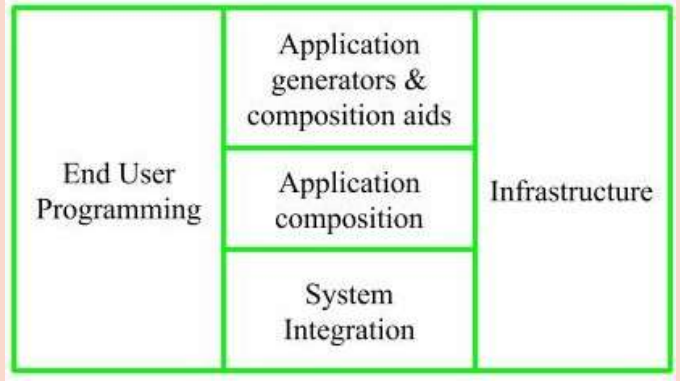
* All defects and errors are categorized by origin.
* Cost to correct each error or defect is recorded

## Fishbone Diagram

* used for managing risks
* helps in diagnosing the data
* backbone

## COCOMO 2 Model (Constructive Cost Model) (also read from PPT and also read this)

* It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.



* End User Programming :
  + End user write the code

## Risk Identification – Types of risk

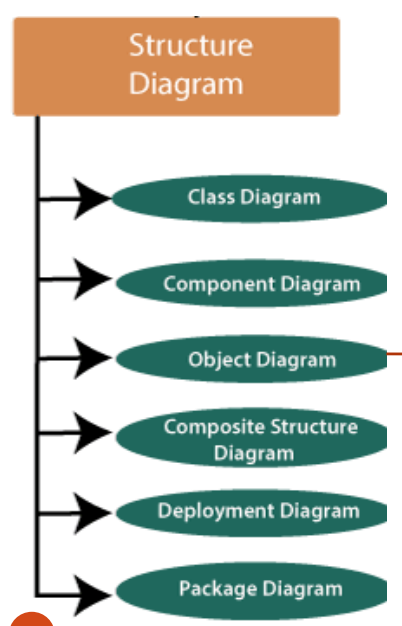
* Technical risks
* Project risks
* Business risks
  + market risk
  + strategic risk
  + budget risk
  + management risk
* Known risks
* Predictable risks
* Unpredictable risks

## Risk Mitigation, Monitoring, and Management (RMMM) plan

* RMMM means risk mitigation (reducing risk of loss from the occurrence of any undesirable event) , monitoring and management plan
* First need to watch on risk then find reason behinds the risk and then save the values for risk !!

# Unit 2 UML

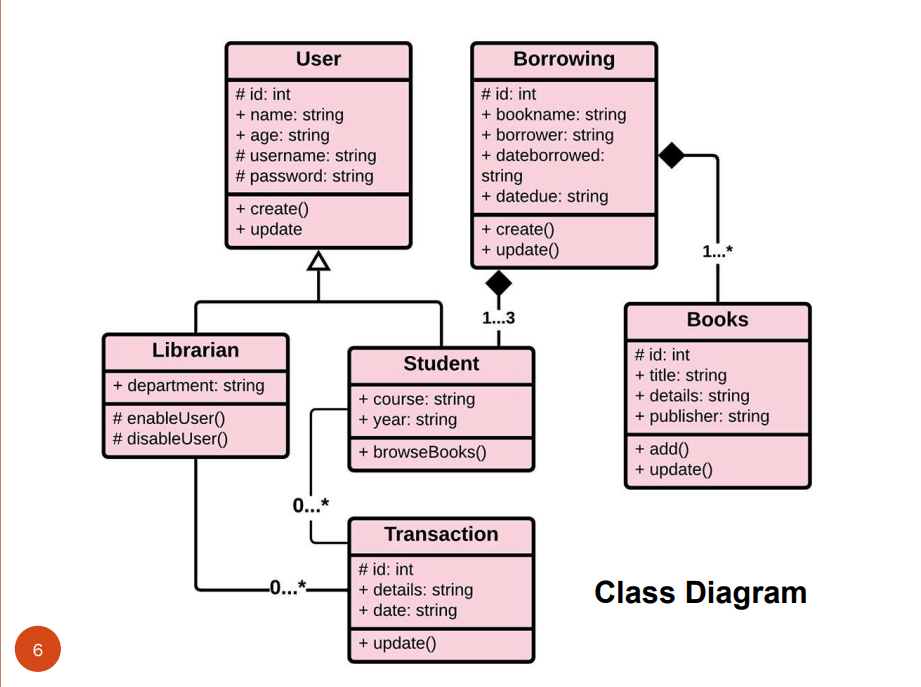
* Full form - Unified Modelling Language
* **Definition** - A UML diagram is a diagram based on the UML (Unified Modeling Language) with **the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes**, in order to better understand, alter, maintain, or document information about the system
* **especially use for OOPS concepts**

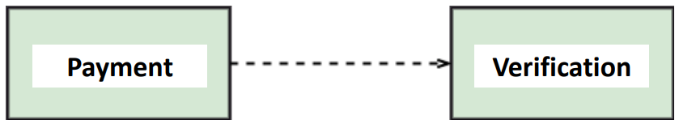


## structure diagrams

1. **Class Diagram**

* It shows a set of classes, interfaces, and collaborations and their relationships, typically, found in modelling object - oriented systems.



* static view of an application
* **Components of class diagram**
  + Classname – capital every first letter of the class
  + Attribute - public (+), private (-), protected (#), and package (~)
  + Methods - . It demonstrates how a class interacts with data.
* **Relationships in Class Diagram**
  + **Dependency:** A dependency is a semantic relationship between two or more classes where a change in one class cause changes in another class. It forms a weaker relationship.
  + **Generalization**: A generalization is a relationship between a parent class (superclass) and a child class (subclass). In this, the child class is inherited from the parent class.
  + **Association**: It describes a static or physical connection between two or more objects. It depicts how many objects are there in the relationship.
  + **Multiplicity in Association** : It defines a specific range of allowable instances of attributes. In case if a range is not specified, one is considered as a default multiplicity.
    - **Aggregation**: An aggregation is a subset of association, which represents “has a” relationship. It is more specific then association.

**Ie.** It defines a part-whole or part-of relationship. In this kind of relationship, the child class can exist independently of its parent class. The company encompasses a number of employees, and even if one employee resigns, the company still exists

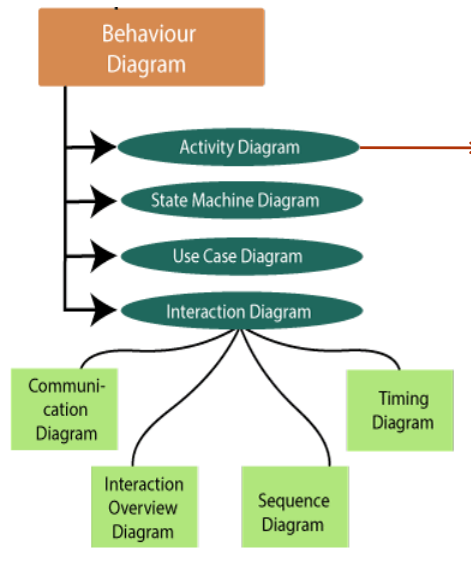
* + - **Composition :** The composition is a subset of aggregation. It portrays the dependency between the parent and its child, which means if one part is deleted, then the other part also gets discarded. It represents a whole-part relationship.
  + **ABOVE POINTS EXAMPLES PLEASE READ FROM ppt ALSO**

1. Component Diagram

* It shows a set of components and their relationships that illustrates the static implementation view of a system.

1. Object Diagram

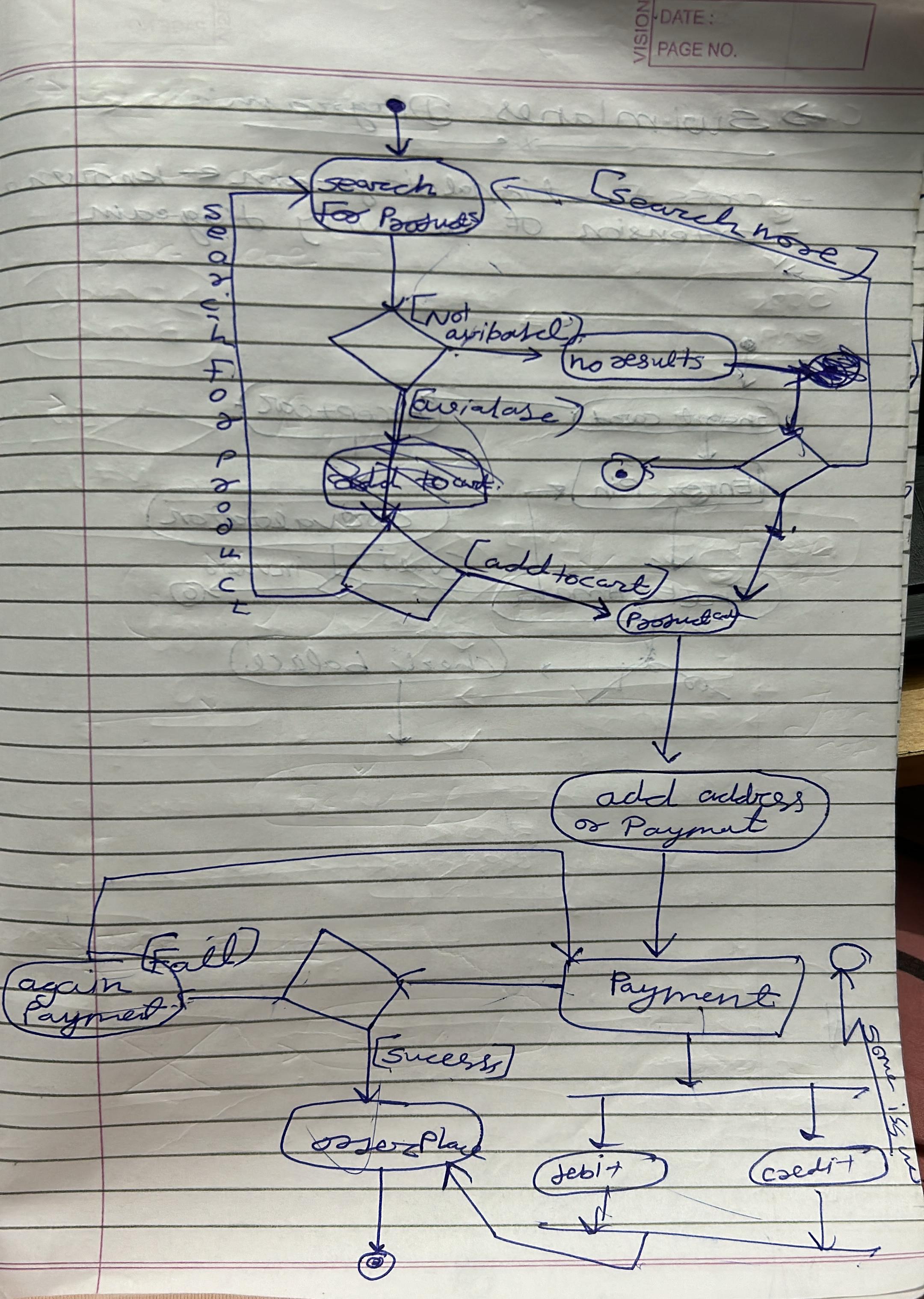
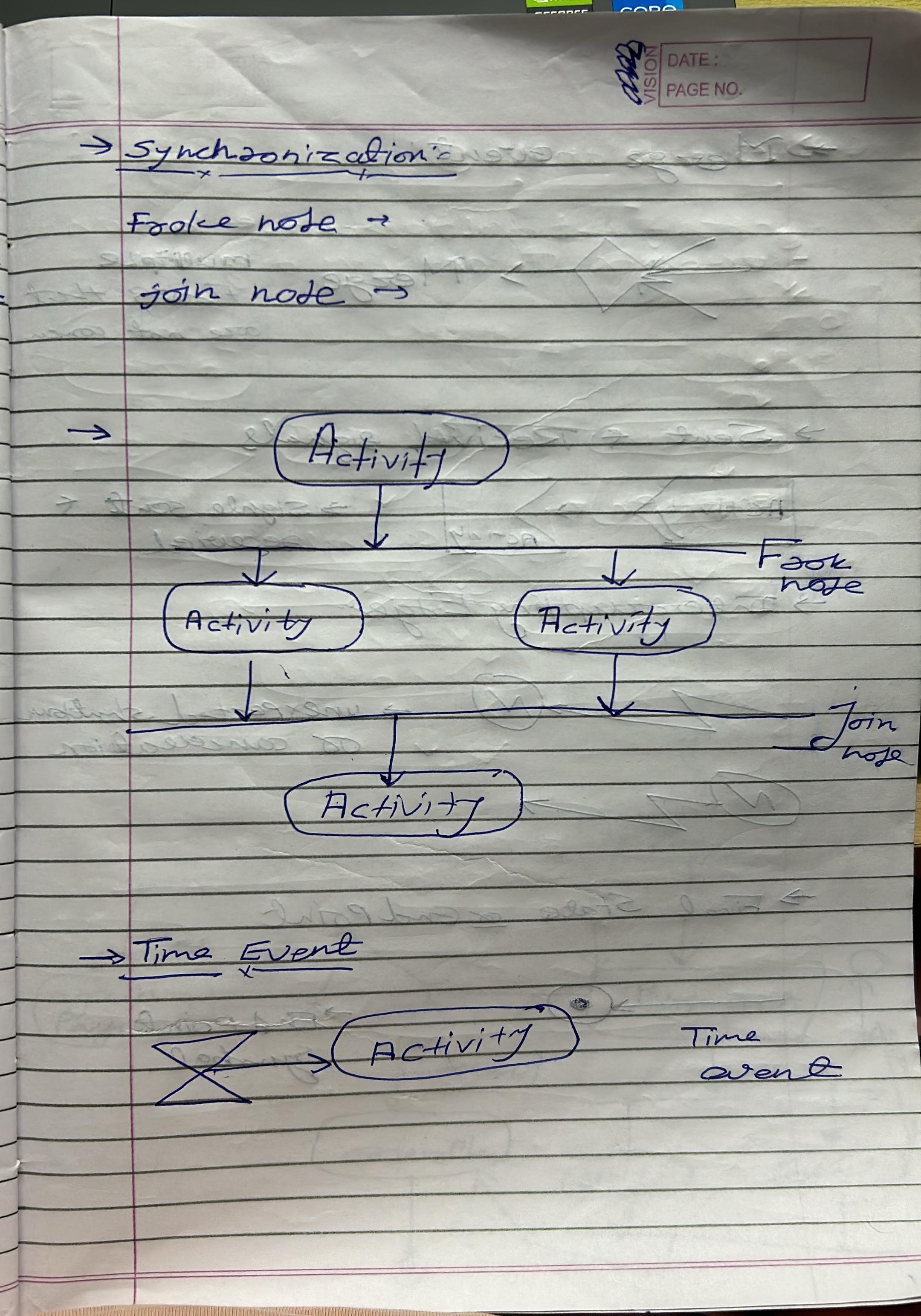
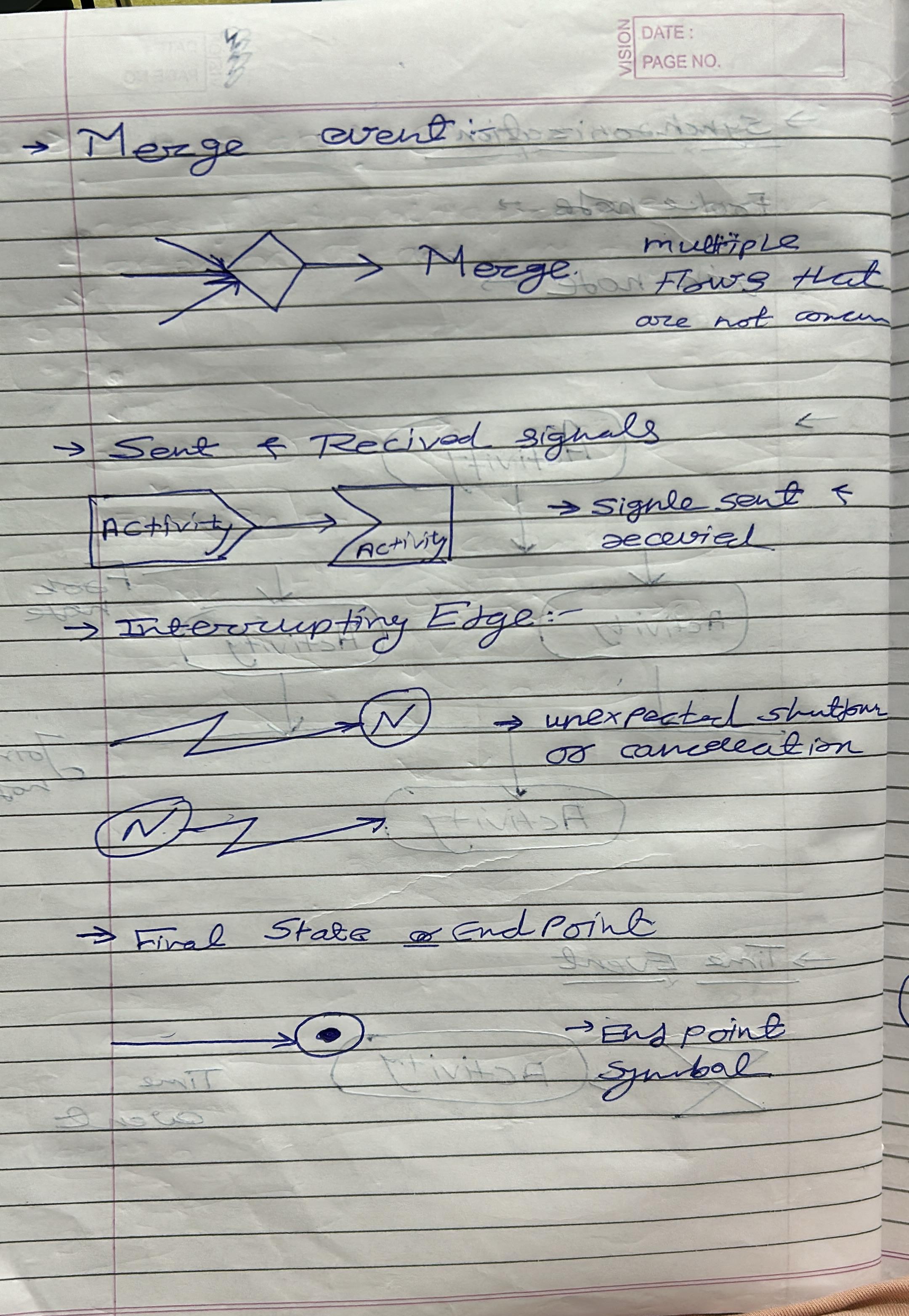
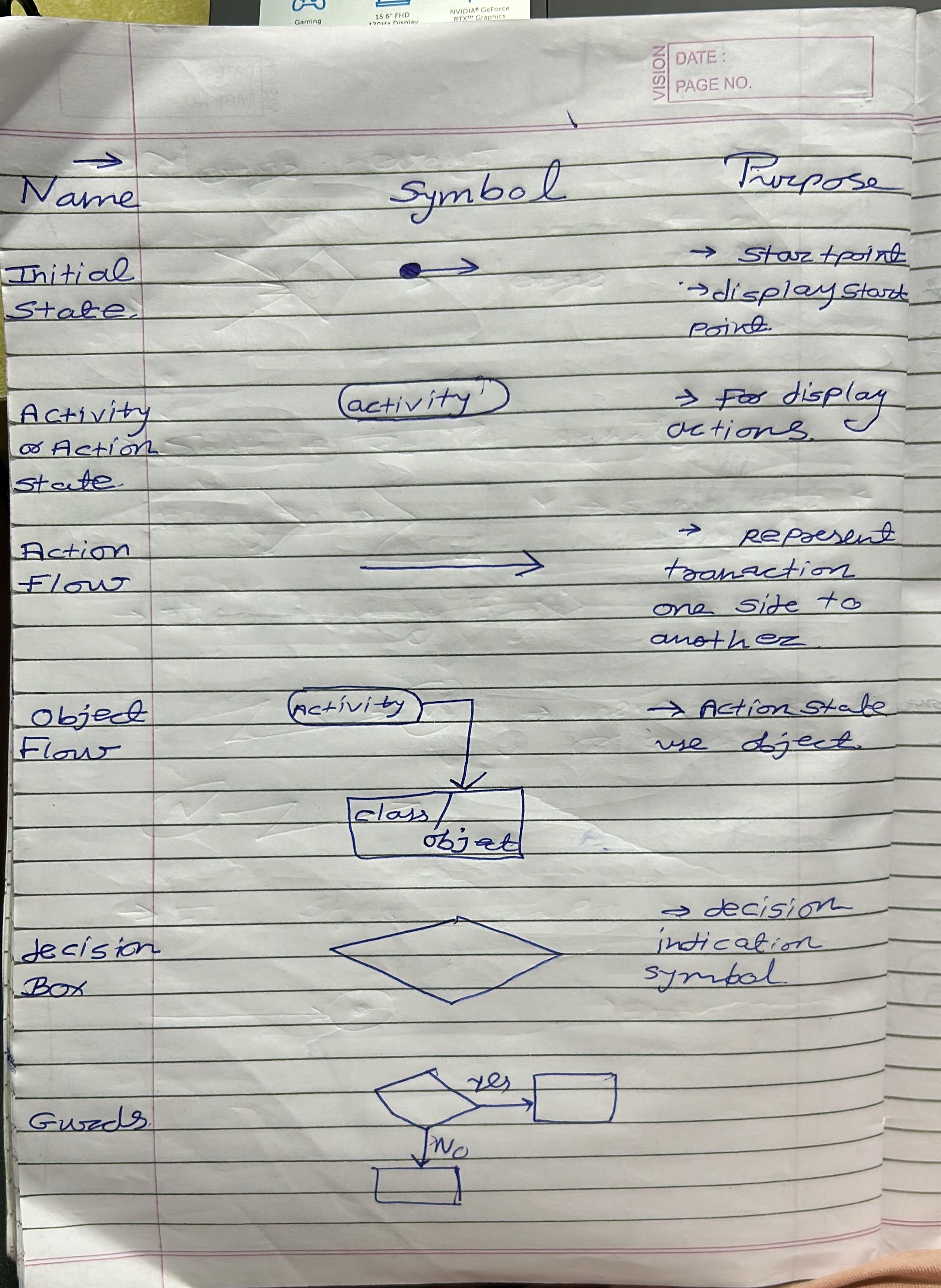
* It shows a set of objects and their relationships, which is the static snapshots of instances of the things found in class diagrams . ( static view at a particular instance of time .

****

**Activity Diagram**

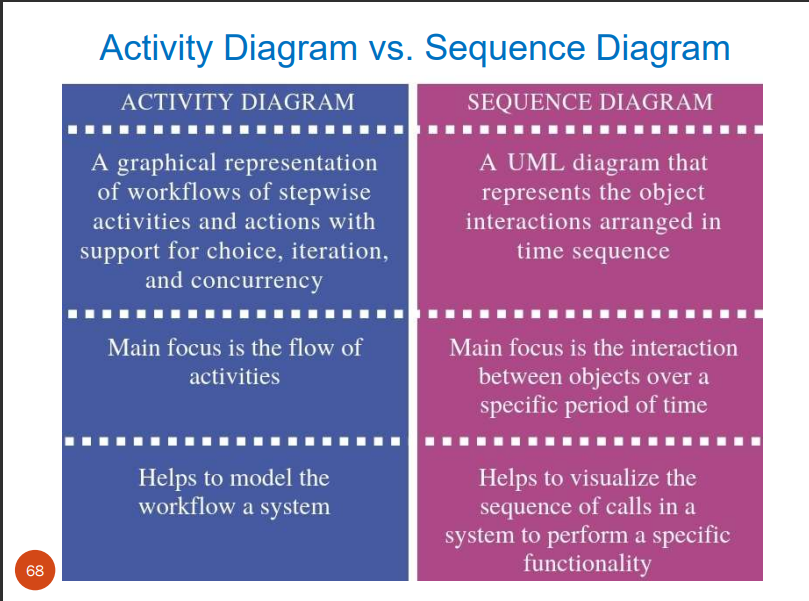
An activity diagram visually presents **a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. But they are not flow charts**

**PLEASE CHECK NOTES FOR SYMBOLS !!**

****

**Sequence Diagram**

* A sequence diagram illustrates the sequence of messages between objects in an interaction . A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.



# Unit – 3 Web Engineering

## Web attributes

Web Engineering (WE) applies sound scientific, engineering, and management principles and disciplined and systematic approaches to the successful development, deployment, and maintenance of high-quality Web-based systems and applications.

## Web Attributes

* Network intensiveness
* Concurrency
* Unpredictable load
* Performance
* Availability
* Data driven
* Content sensitive
* Continues evolution
* Immediacy
* Security
* Aesthetics

## Web Testing

* Link testing
* Browser testing
* Load testing
* Stress testing
* Continues testing
* Security testing

# Unit 4: Software Architecture and Design Patterns

* It is a pictorial representation of the IT System
* Blueprint of the system

## Goal of software architecture

* to identify requirements that affect the structure of the application.

## Principles of Software Architecture

* SOLID – each character define the one principal of software architecture.

## Different Software Architecture pattern

* Layered Pattern
  + components(code) in this pattern are separated into layers of subtasks and they are arranged one above another
  + known as ‘N-tier architecture’. Basically, this pattern has 4 layers:-
    - Presentation layer
    - Business layer
    - Application layer
    - Data layer
  + Idel for online shopping e commerce websites like amazon and flipkart
* Client Server Pattern
  + One server and multiple client
  + Use for file , data and other things
  + i.e. email,www,filesharing app etc
* Event Driven Pattern
  + Event-Driven Architecture is an agile approach in which services (operations) of the software are triggered by events
* Pipe Filter Pattern
  + Pipe and Filter is another architectural pattern, which has independent entities called filters (components) which perform transformations on data and process the input they receive, and pipes, which serve as connectors for the stream of data being transformed, each connected to the next component in the pipeline
* Peer-to-Peer Pattern
  + individual components are known as peers
  + peers acts as a server or client also
* Model-View-Controller Pattern

