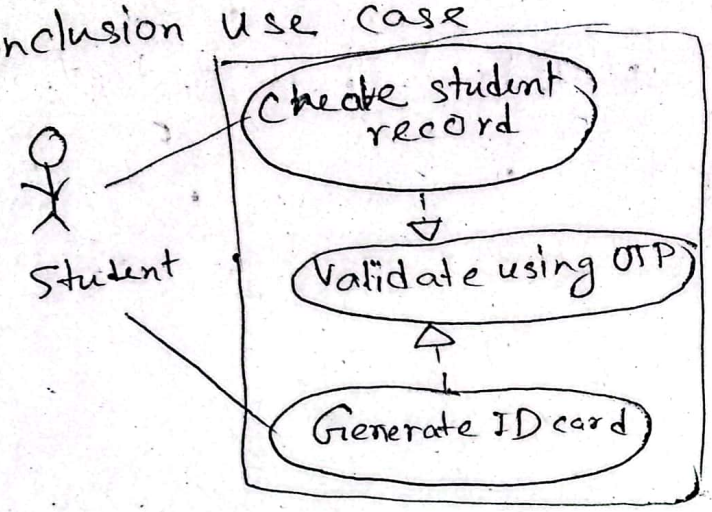
Use Case

captures the functionality of system from user perspective

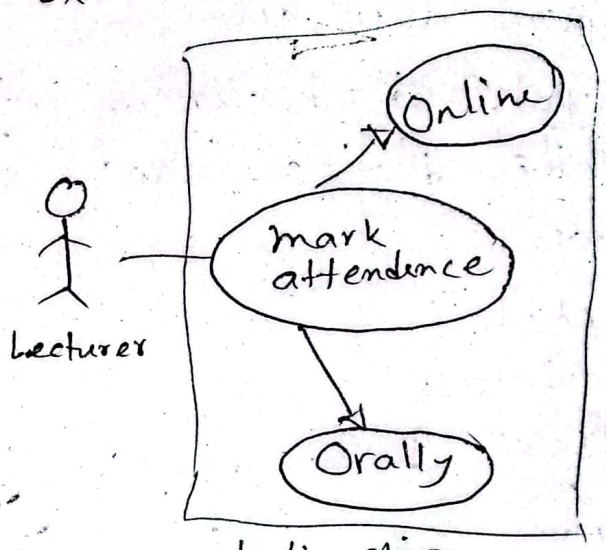
Include Relationship

Inclusion use case



Extend Relationship

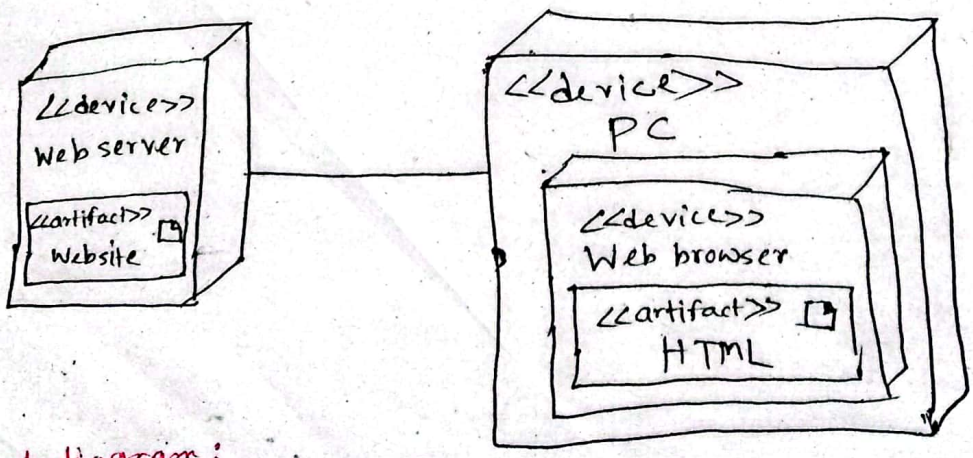
Extension use case



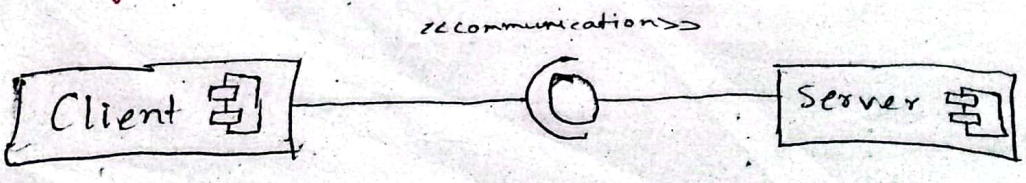
Object diagram: Snapshot of objects and relationship between them.

- **Component diagram:** Physical aspects of system and relationship between them.

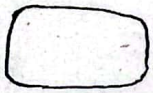
Deployment diagram: Shows how software and hardware components are distributed across a network or system.



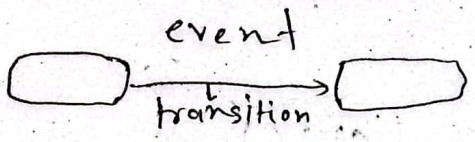
Component diagram:



State Diagram: Dynamic aspects of the system is captured by state diagram

State 

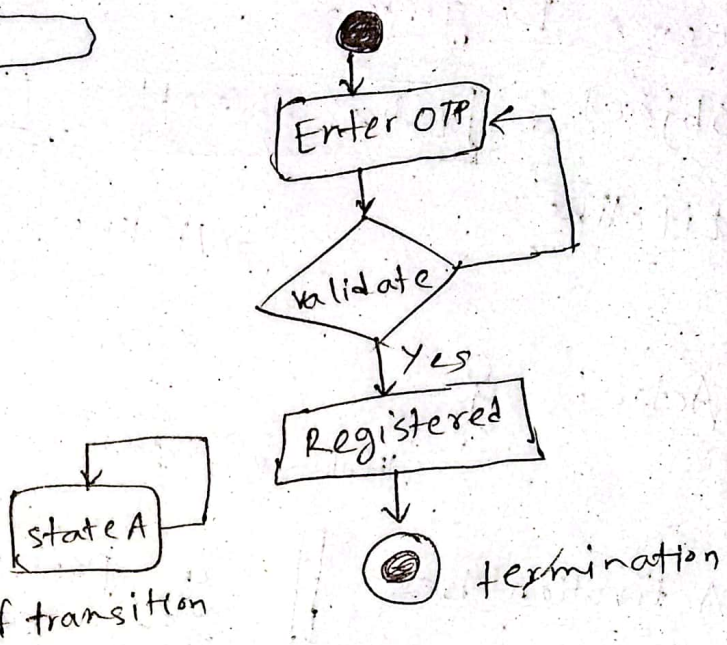
Event : এক state থেকে অন্য state এ যায় কেন.

transition: 

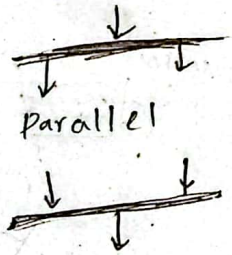
initial state : 

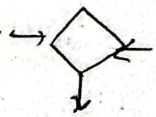
Final/End State : 

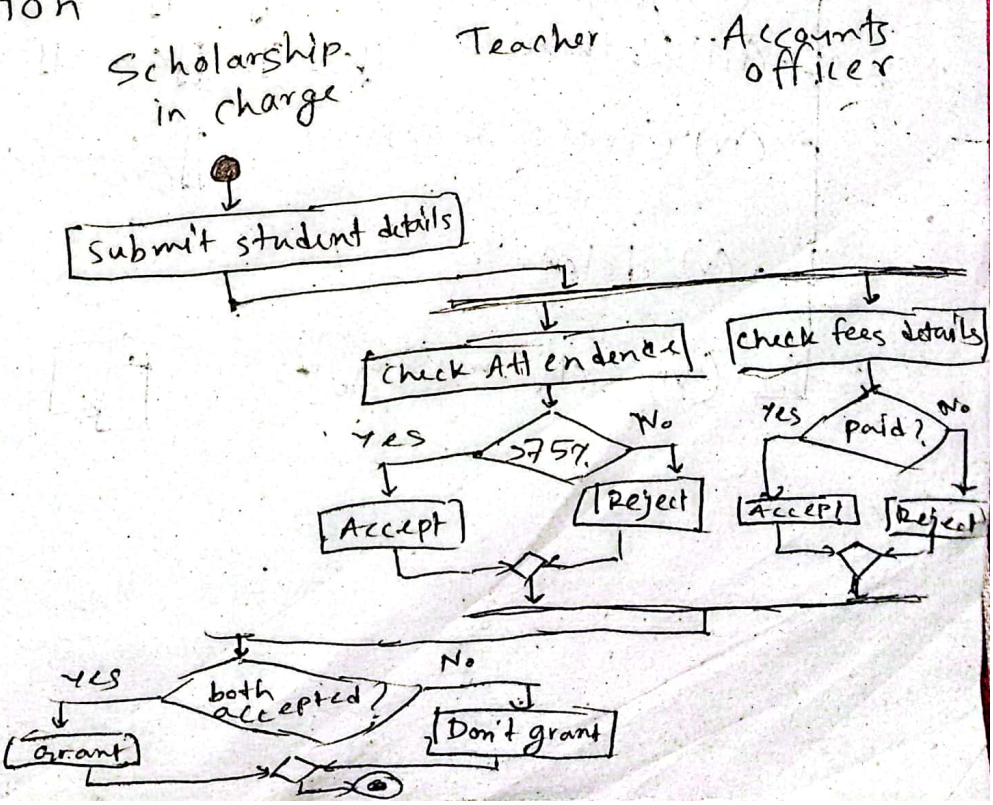
Decision Box : 



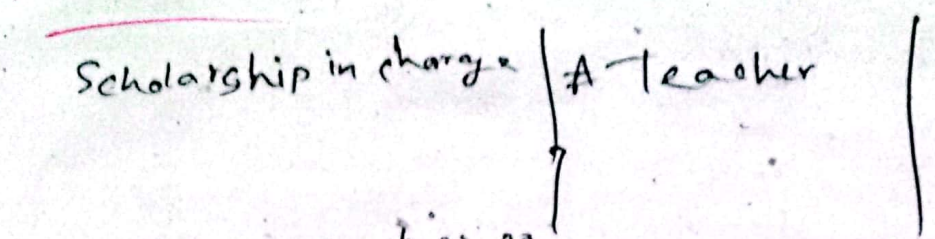
Activity Diagram: Like state diagram Except most transitions are caused by internal events, such as the completion of a computation







Swimlanes :



swimline দিয়ে class এর activity আলাদা করা হয়েছে

Sequence Diagram:

object : Esha : Student 01 : C1

Lifeline : → Lifeline

Actor : → lifeline of actor

Activation Bar — a period or duration for which your respective object is active

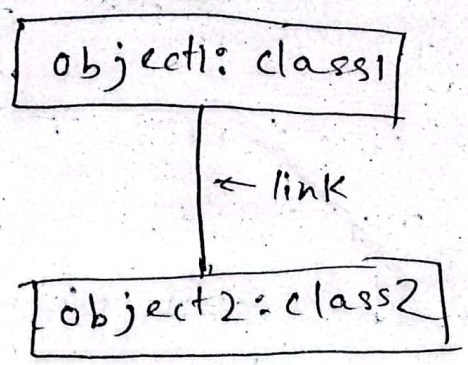
message

- (i) Synchronous — wait করে receiver থেকে reply আসার জন্য
- (ii) Asynchronous — wait করেনা
- (iii) Return
- (iv) Create message — «create» → 02 : C2
- (v) delete : → X no lifeline — Object delete
- (vi) Self message

Communication diagram

object :

link :



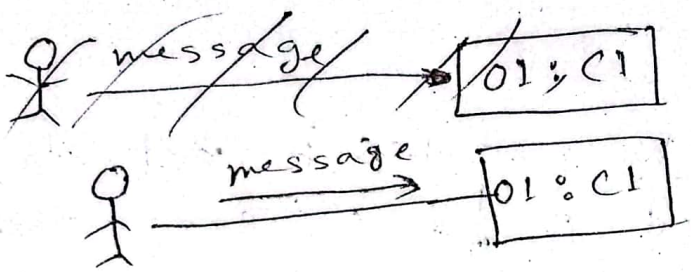
1 : msg →

2 : msg →

3 : msg →

4 : msg →

~~mess~~ actor :



Coupling: measures the degree of interdependence between software components.

Data coupling ↑ best

Stamp coupling

Control coupling

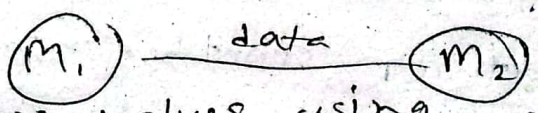
External coupling

Common coupling

Content coupling worst

Data module: Two modules communicate with each other only by passing data.

cash হাত দিলে internal কোনো ব্যাপার নাই,
bank, credit card G problem মানেও m2
এত সোজা নাই কিছু আছে ব্যাপার



Addition of two values using call by value.

Stamp Coupling: Two modules communicate with each other by passing of data structure.

m_2 টাকা চাইল m_1 purse থেকে নিতে বলে,
টাকা কম পেলে m_2 কে যন্ত্রণা করবে,

addition of two values by ~~call~~ using call by reference

Control Coupling: Two modules ~~is set to be controlled coupted~~ if ~~they~~ communicate using control information with each other.

e.g.; dependence of two modules on each other because of flag

signal green না হলে train চলবে না,

External Coupling: Measure of dependency between a software system and external entities like libraries or services.

যদি train green flag, track যাবে

Common Coupling: If two modules share some global data, eg. synchronization issue between the processes

Content Coupling: When one module is a part or context of another module

Routine Call Coupling: Occurs when one routine calls another.

Type Use Coupling: Occurs when a module uses a 'data type' defined in another module.

Inclusion or Import coupling: ~~When~~ Occurs when one component imports a package or one component includes another

---

Cohesion: Measure of functions strength of the module.

Function
Sequence / Layer
Communicational
Procedural
Temporal
Logical
Coincidental

high

low

Coincidental Cohesion: Only relationship between the functions in a module is random or coincidental.

Logical Cohesion: All the elements of a module perform similar or slightly similar operations. mouse, printer, scanner functions are written in the same module

police, army [illegible] training

Temporal Cohesion: When elements within a module are grouped together because they are executed at the same time or share a common temporal context.

e.g: User authentication function for login, logout and session expiration are grouped together due to their related timing durig user interactions.

Procedural Cohesion: Functions of a module are related to each other through a flow control.

File handling
- opening
- reading
- writing
- closing

Communicational Cohesion: Different functions are operating on the same data structure
push, pop

Sequential Cohesion: When elements in a module are grouped together because they are executed in a specific sequence to perform a single task or procedure.

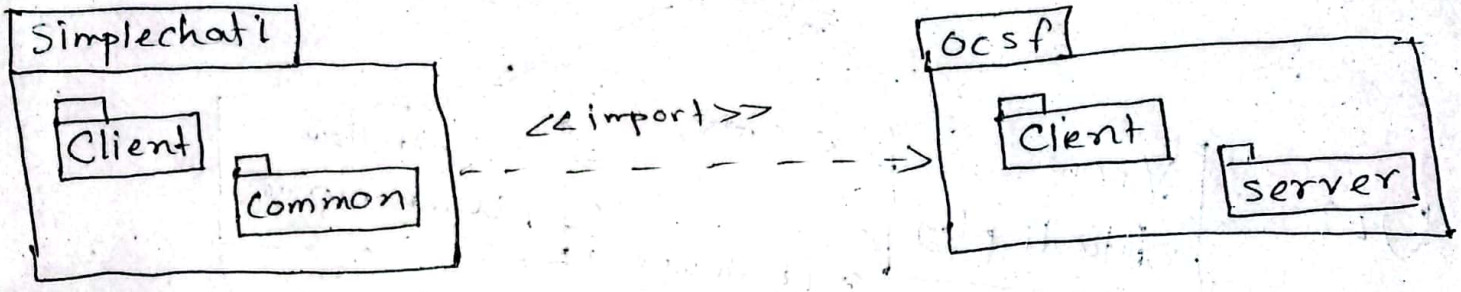
ex: same as temporal

Functional Cohesion: Different functions of a module cooperate with each other to perform a single functional.

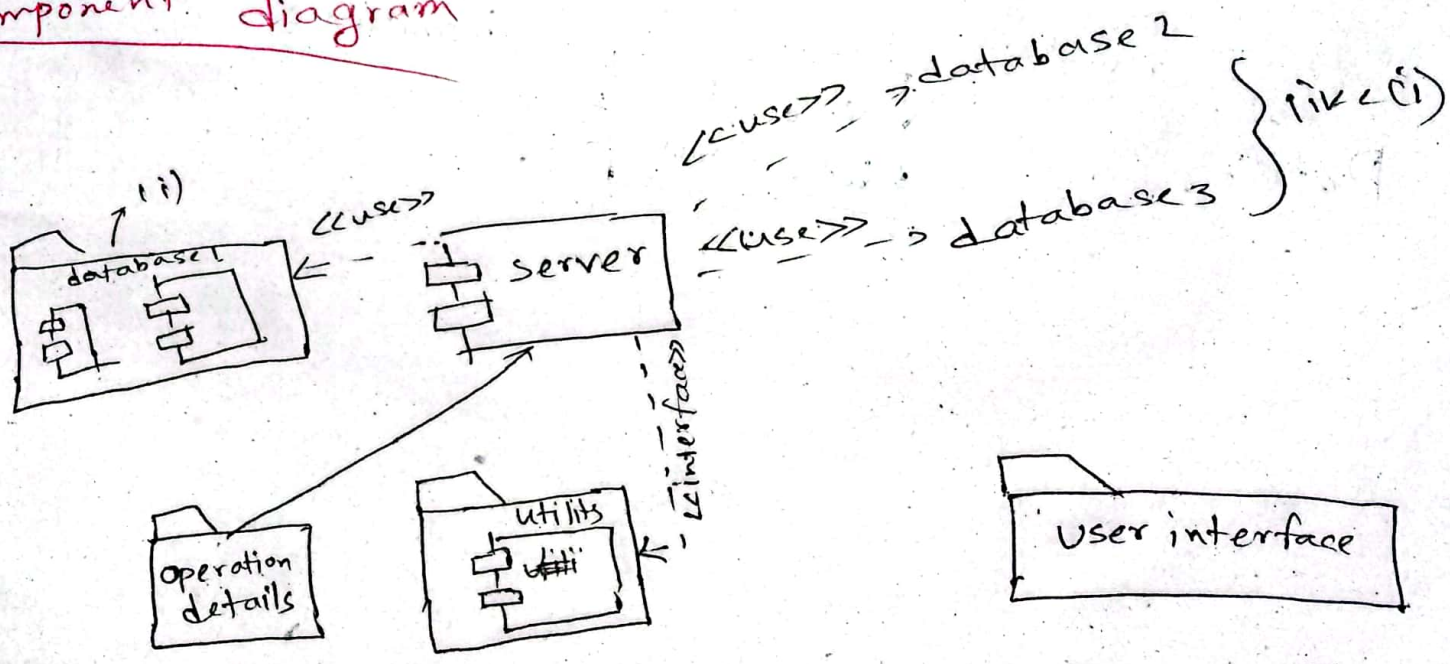
~~data management~~
- adding
- updating
- deleting
- retrieving info

} data management

Package Diagram:



Component diagram



MVC: to manage code. Not language dependent.