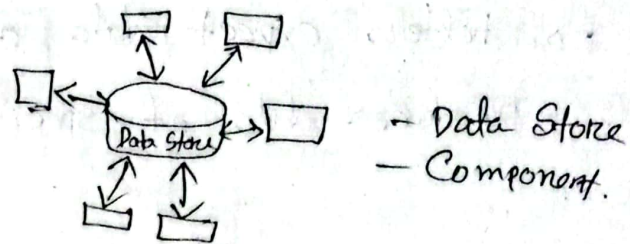


- (4) 3rd pattern is 3rd, 3rd 3rd, type 3rd + Note (10)
- Slide (8-10) for theory

Data Centered Architecture : A data store (e.g. a file or database) resides at the center of this architecture and is accessed frequently by other components that update, add, delete or otherwise modify data within the data store.



It has two variations.

1) The data store is passive. That is, components access the data independent of any changes to the data or the actions of other component.

2) A variation on this approach transfer

2) Blackboard: Data Center sends notification to components when data of interest changes.

Benefits :

1) ~~less~~ Decouples components / clients. Any components can be added/removed independently.

2) Facilitates communication between clients when data changes. → Message passing

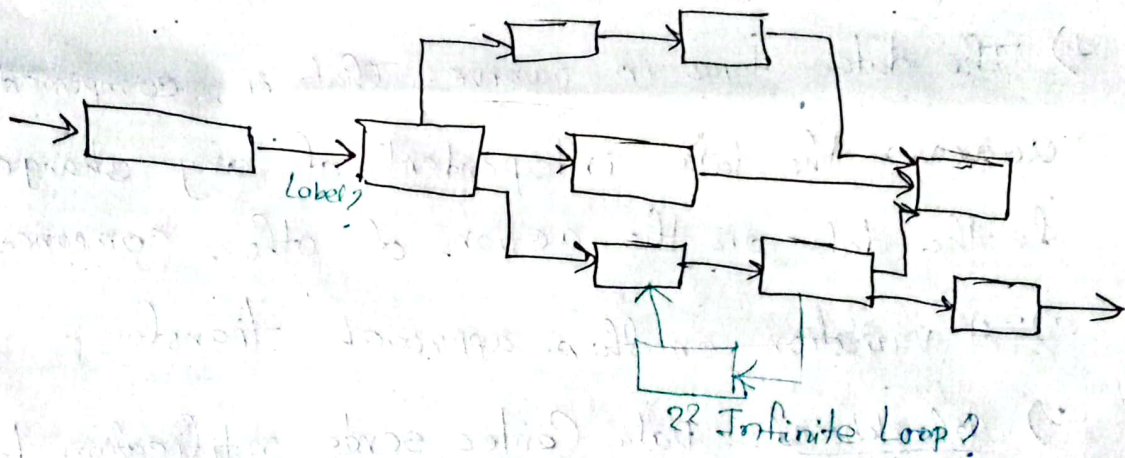
Disadvantages :

1) Centered so single point of failure.
2)

Data Flow Architectures: A pipe-and-filter pattern has a set of components called filters connected by pipes that transmit data from one filter to the next.

→ Each filter works independently, irrespective of filters upstream & downstream.

→ Filters expect data of certain form and produces data of specific form.



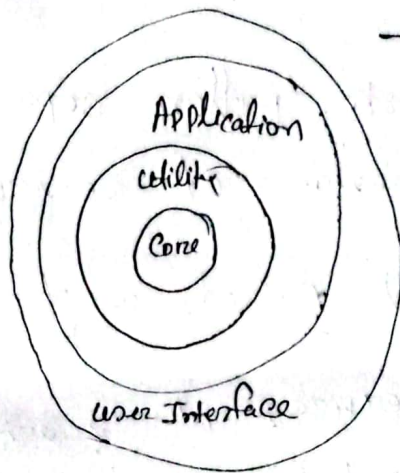
Batch Sequential: If the data degenerates into a single line of transform, it is termed batch sequential. It accepts a batch data and then applies a series of sequential filters to it.

Benefit

Disadvantage:

Layered Architectures : In this architecture, a number of different layers are defined, each progressively closer to machine instruction set.

- Outer layer components service UI operations.
- At inner layer, components perform OS interface.
- Intermediate layer provide utility services & app business logic.

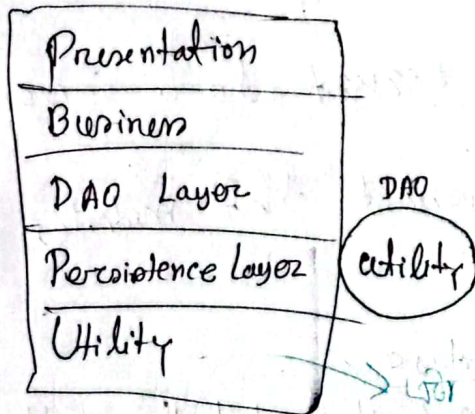


→ ONE WAY

262

Layers

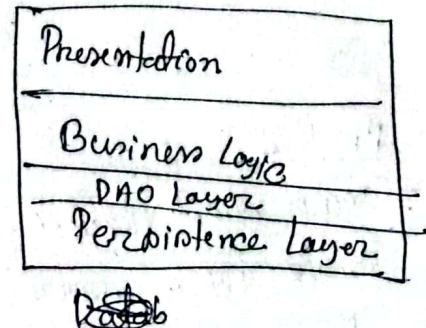
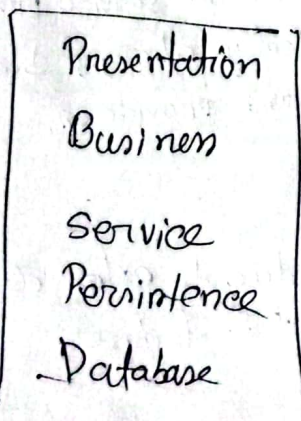
- 1) Presentation/UI
- 2) Application/Business Logic
- 3) Domain/Model Layer
- 4) Data Access Layer/Infrastructure Layer
- 5) Persistence Layer/Data Layer



DAO

Utility

→ User Interface



Disadvantage

- Increased cost & Complexity
- Performance penalty

Advantage

- 1) Better maintainance
- 2)

✓ The layered pattern defines layers and a unidirectional allowed-to-use relation among layers.

☐ Broker Pattern: The broker pattern separates users of services from providers of services by providing an intermediary called a broker.

- When a client needs service, it queries a broker (via a service interface.)
- The broker forwards the request to a server.
- Broker is a runtime component that mediates communication

Broker

- accept client request
- locate approx server
- Forwards request
- Return result to server

Advantage

- Server's don't need to know the nature & location of service providers
- Easy to dynamically change bindings between users and providers

~~- Scalable~~

Disadvantage

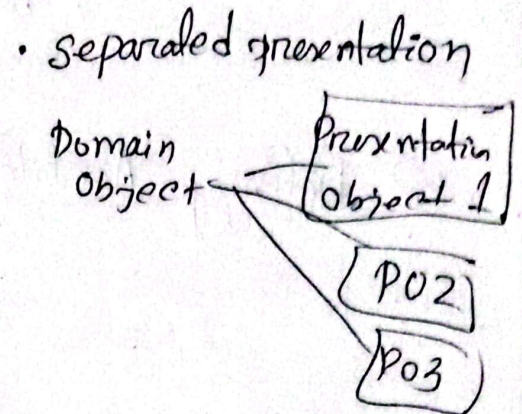
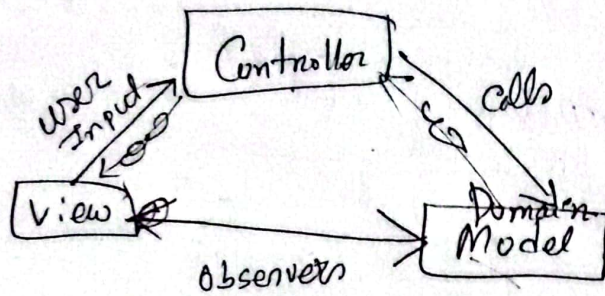
1. Latency due to extra element
2. Single point of failure

~~3. etc~~

3. Brokers can become complex.
4. Hard to test.
5. Target for security attacks.
6. Scalable w/ dec to added complexity.

need figure.

MVC:



→ Controller's job is to take user input and figure out what to do with it.

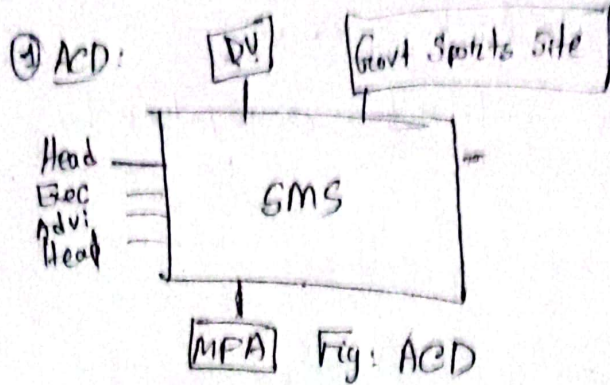
→ All views & controllers observe the model.

→ Multiple screen

→ Unidirectional so separation of concern

Practice

LEM: Sports Management System

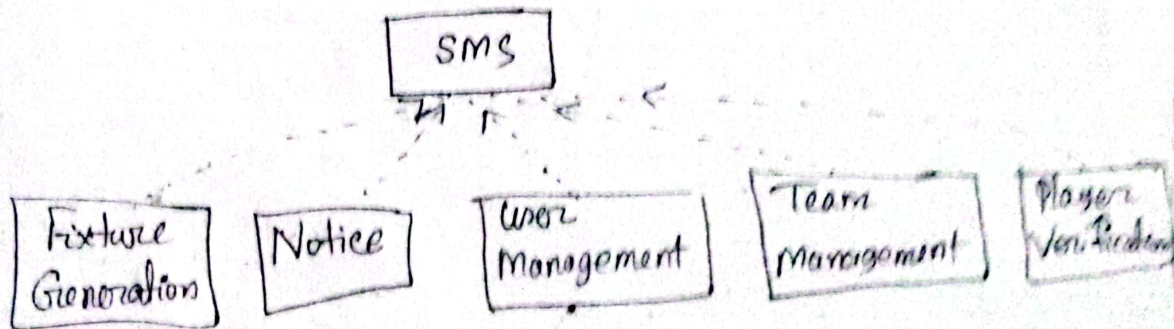


MVP
Landscape
Markers
Wpe

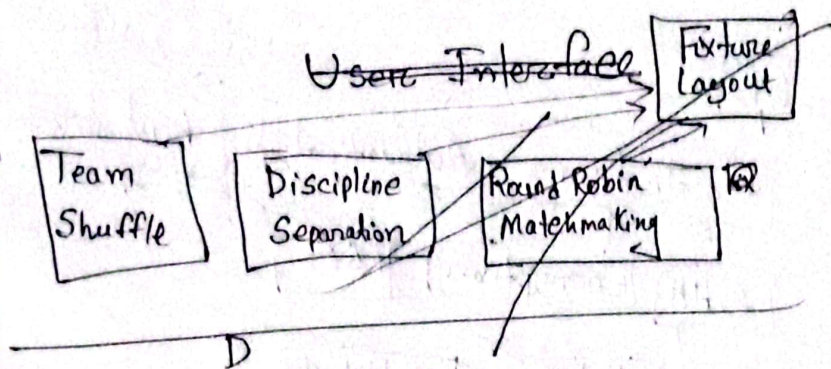
② Defining Architectural Patterns

③ Refining the architecture into components

List: Fixture Generation, ^{User management} Account Management, ^{Authentication} Authentication, ^{Notification, Notices} Notification, Notices, Team Management, ^{Player Verification} Player Verification



Fixture Generation : Random Number generation, Shuffle
 Team Shuffle, Separation of Discipline, Round Robin Matchmaking
 of Teams
 Fixture layout generation.



pipe & Filter X
 MVC X
 Layered.
 Broker X
 OOP X
 Data Centered X
 Main Program X

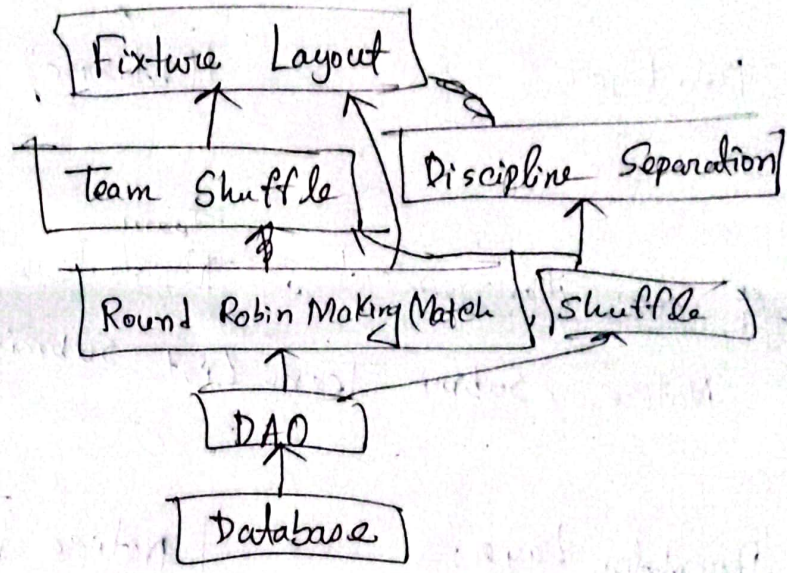
Presentation Layer:

Business Logic Layer:

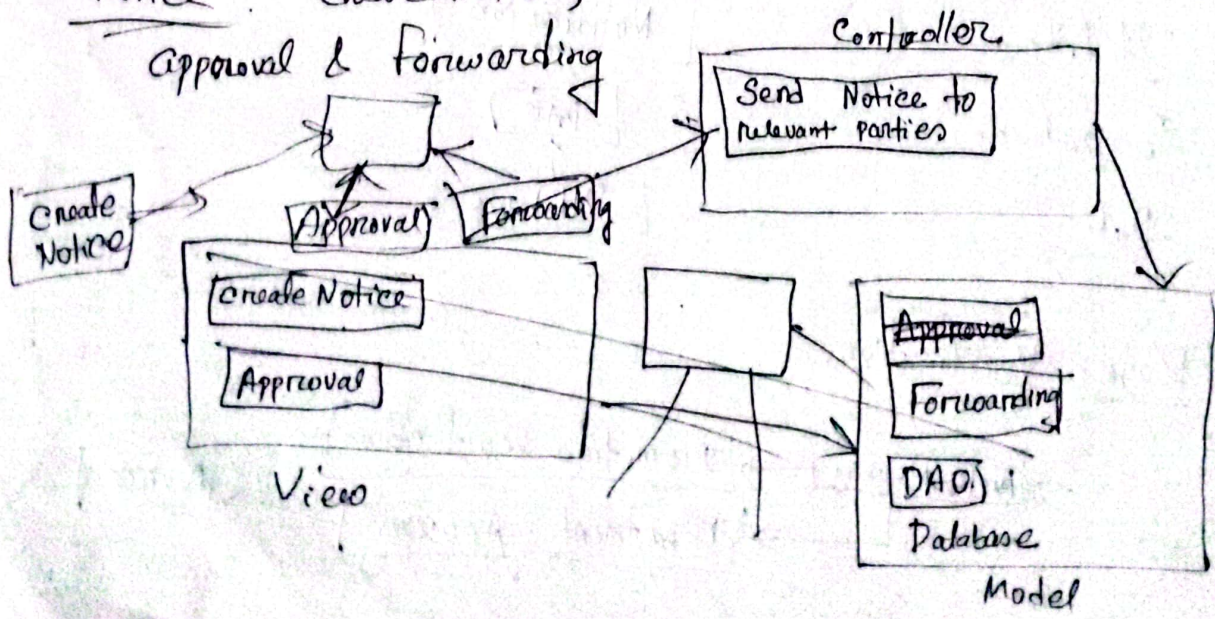
Utility Layer:

Persistence Layer:

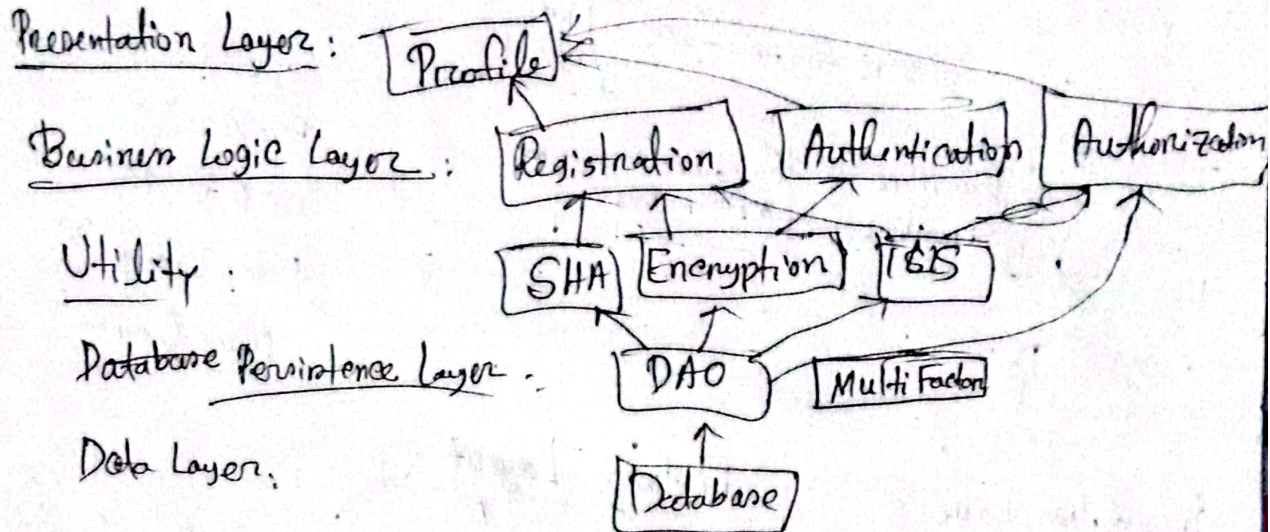
Data Layer :



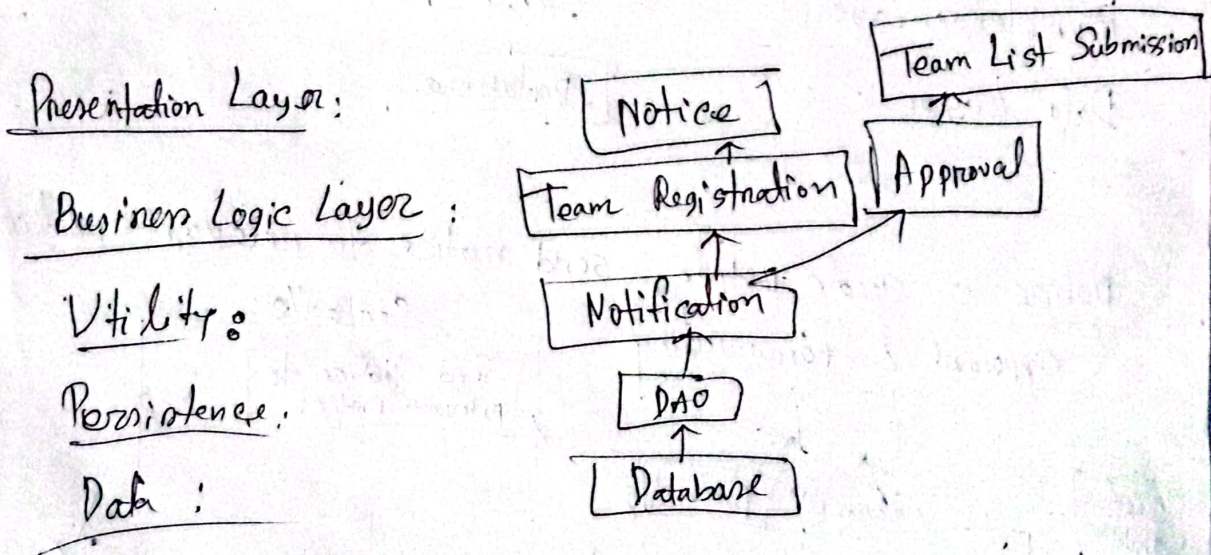
Notice : create Notice, send notice to relevant parties
 Approval & Forwarding



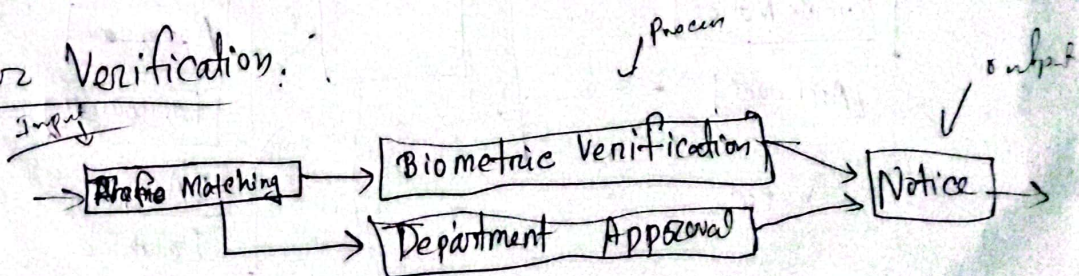
User management: Authentication, Authorization, registration, profile, MFA



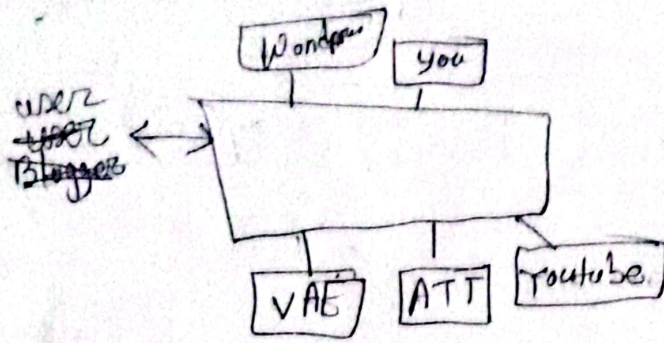
Team Management: Team Registrations, Approval, Notice
Notice, Submit Team List Submission



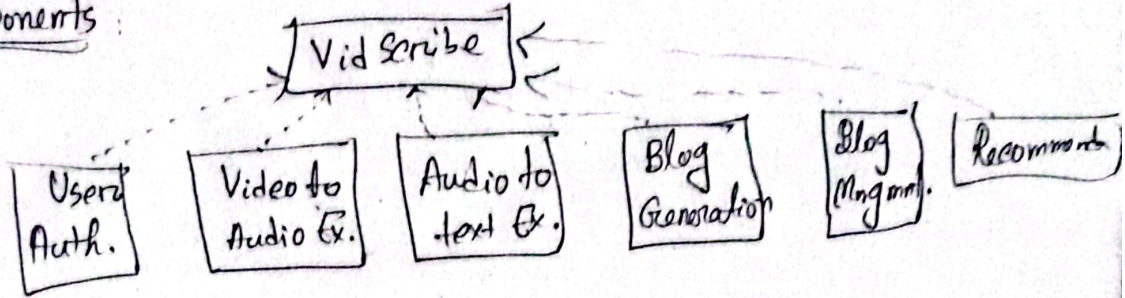
Player Verification:



Vid Scribe :



Components :



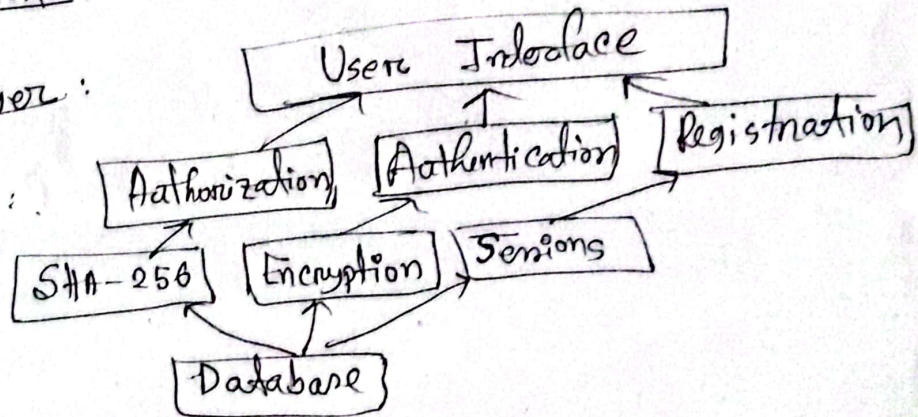
User Authentication :

Presentation Layer :

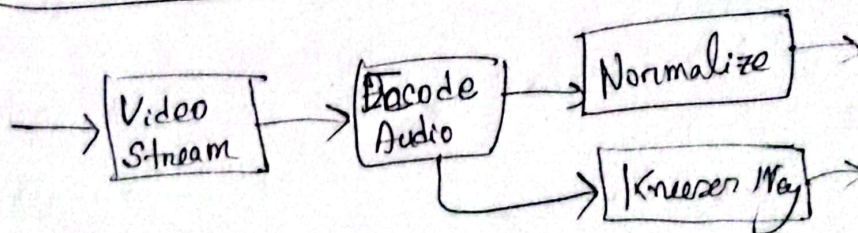
Business logic :

Utility :

Data Layer :



Video to Audio Extraction :



Audio to text

