

Chapter-4

Date _____
Page _____

Web Application Architecture

Application

- # Definition of web Architecture = (WAA)
- Web application architecture defines the interaction between application, middleware system and databases to ensure multiple application can work together. (concurrency betw application)
- When the user types a URL and hits 'go', the browser will find the where the website lives on and request that particular page.
- Web app WAA represents relationship b/w ~~such~~ ^{other} user interface, transaction process, database, and ~~so~~
- The primary objective is to ensure that all elements work together correctly.

(WAA)

- # Basic Components of web Application Architecture :-
- There are two basic components of WAA. They are:-
- ① User Interface Components
 - ② Structure Components
- Structure are also divided into client side and Server side.
- (1) User Interface Components :-
- User In= Web application architecture components of

the user interface consists of all the interface element like activity log, settings, dashboard, notification, etc.

→ They are a part of web application interface layout.

② Structural Components :-

→ ~~Web~~ Web application structure ^{includes} consists of components of consisting of client side and server side.

→ ^{Client} The components is created with javascript, HTML or CSS, etc.

→ Web browsers run the code and convert it into the interface, so there is no need for an OS adjustment.

→ As for the server component, its built with Java, .Net etc.

→ The Server consists of two parts :- Applogic and database.

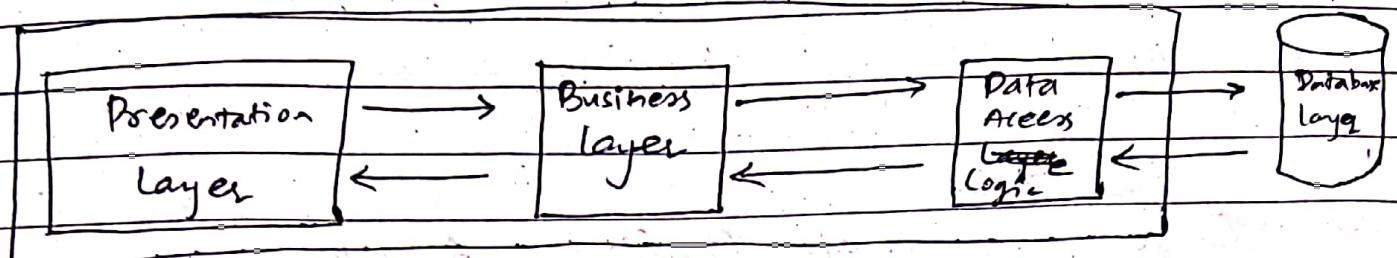
N-tier Architecture:-

- There 3 types of N-tier Architecture :- They are :-
- ① 1-tier Architecture
 - ② 2-tier Architecture
 - ③ 3-tier Architecture

(See more details in note
EAD-Combined.pdf Pg - 246)

Date _____
Page _____

① 1-tier Architecture :

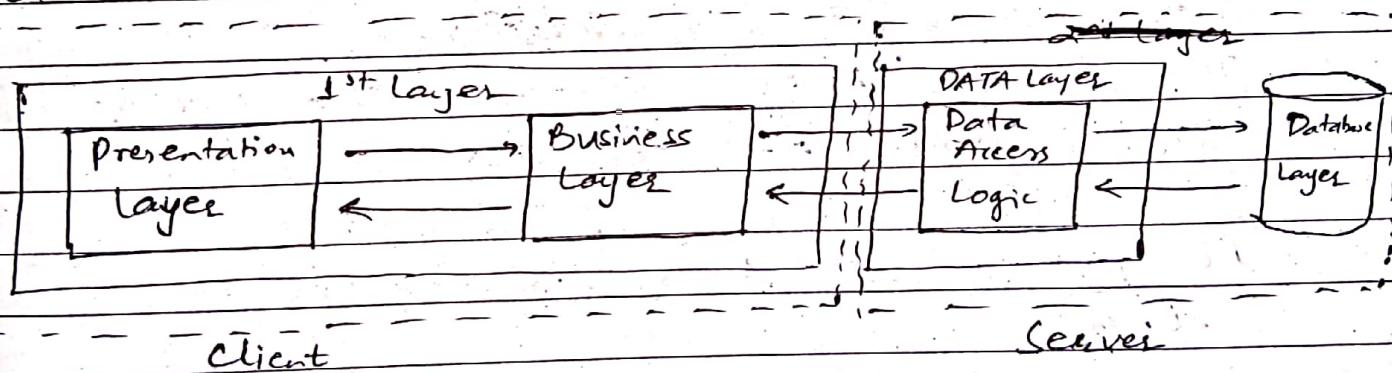


→ Presentation layer, Business layer, Data layer are tightly connected.

Scalability, Portability and Maintenance are difficult

Issue

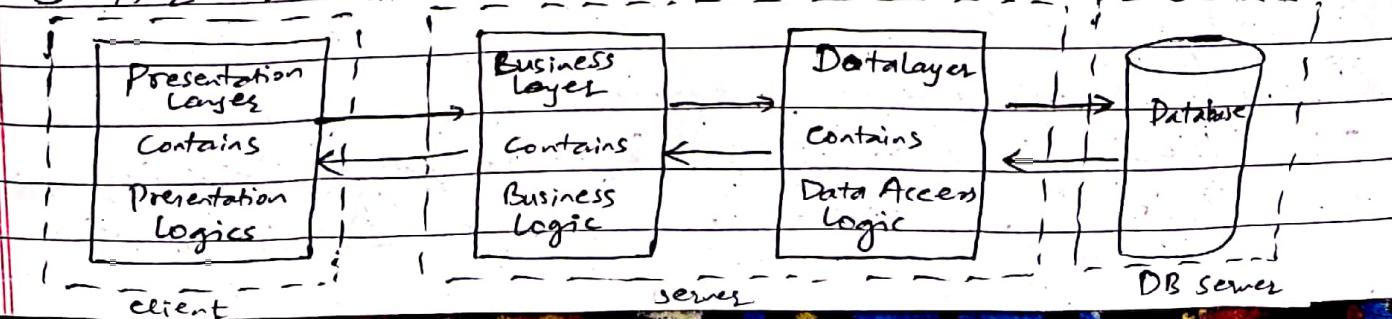
② 2-tier Architecture :



→ Database runs on Server

→ Presentation and logic layer still tightly connected.

③ 3-tier Architecture :



→ Presentation layer, Data layers disconnected.

So, scalability, portability and maintainability is simple.

2

→ Layers of Web Application Architecture =

① Presentation Layer

② Business Logic Layer / Domain Layer

③ Data Service Layer

④ Data Access Layer

① Presentation Layer : (PL)

→ PL displays the user interface and makes user interface more straightforward.

→ This layer has UI components that process and displays data for users.

→ PL provides all the required data for the client slides.

→ The primary goals of the PL is to get input data, process user requests, send them to data services and provide the results.

→ It's available through a browser and contains UI elements that cooperate with the system layer.

② Business logic layer : (BLL)

→ BLL is responsible for the proper data exchange.

→ This layer defines the logic of business operations and rules.

- Logging in the website is an example of a business logic layer.
- Two more names for this layer are domain logic or Application Layer, which completes the processing of customers queries from the browser and discrete the ways of access to his data.

③ Data service layer:- (DSL)

- DSL of web-based application architecture transmit data processed by the business logic layer to the presentation layer.
- By transferring the data ^{operation} with the BLL to presentation layer, DSL protects the information of application web-architecture by isolating business logic from the client-side.

④ Data access layer:- (DAL)

- DAL offers simplified access to data stored in persistent storages like binary and XML files.
- Data access layer also manages CRUD operations. Create, Read, Update, Delete
- It is also called the persistence layer storage, which is united with the business layer.

QUESTION

Presentation Layer:-

- ① Presentation layer Components
- ② Presentation layer Approach
- ③ Presentation layer design considerations.

1. Presentation Layer Components:-

→ PL Components are:-

- (a) User Interface (UI) components
- (b) User Process Components

(a) User Interface (UI) Components :-

→ UI components provides a way to interact with a computer application.

→ They renders and format data for user.

→ They acquire and validate data input by user.

(b) User Process Components:-

→ User Process Components synchronize and arrange user interactions.

→ Separate user process components are may be useful if the user interface is complicated.

2. Presentation Layer Approach:-

→ PL Approaches are:-

(a) Determine how we will present data. Choose a data format for PL and to ^{present to} ~~interact with~~ UI present to UI.

(b) Determine data validation strategy. ^{by using} ~~by using~~ system different data validation technique to protect ^{system} ~~from~~ untrusted input.

- ④ Determine business logic strategy
- ④ Determine strategy to communicate with other components.

③ PL Design Considerations.

→ PL Design Considerations are:-

- ① Use relevant relevant pattern:
Use the pattern that ~~so~~ provide solution for common presentation problem.

- ② Design for separation of concern

Use dedicated UI ~~the~~ components that focus on render and display.

- ③ Consider human ~~design~~ interface design guidelines

Review your ~~organizational~~ guidelines for user UI design.

- ④ Attach to design - a user-driven design principle:-

Understand your customer before designing the presentation.

WVime

Business Logic Layer:-

- ① Business logic Components
- ② Business logic Approach
- ③ Business logic Design Consideration.

① Business Logic Components

→ BL Components are:-

- ① Business Components
- ② Business entity Components

③ Business workflow

④ Application facade (optional)

(a) Business Components :-

- After user process collects the ^{required} data, it's the data can be operated by using business logic.
- The rules will describe how the data will be manipulated.
- Rules ~~can~~ be simple or complex depending on the business itself.

(b) Business entity Components :-

- Business entity entities are used to pass the data between other components.
- Data represents real-world business entities, like products and orders.

(c) Business workflow :-

- Many business processes involve multiple steps that must be performed in correct order.
- Business workflow defines multistep business process.

(2) Business logic Approach :-

- BL Approaches are :-

- ① Design Create an overall design for business layer.
- ② Design Business component.
- ③ Design Business entity components
- ④ Design Business workflow components

(a) Create an overall BL design for business layer.

- Identify the consumer in Business layer.
- Determine how we will expose the Business layer.
- ~~Design~~ Determine the security requirements for Business layer.

(b) Design Business Components:-

- Identify the Business Components.
- Choose Transaction Support.
- Identify how business rules are handled.
- Identify patterns that fit the requirements.

(c) Design Business Entity Components:-

- Identify common data format for business ~~entity~~ entities.
- Choose data format.

(d) Design Business Workflow Components:-

- Identify workflow style using Scenarios.
- Determine how business rules are handled.
- Choose a workflow solution.

(e) Business logic Design Consideration:-

- BL for design consideration are as follows:-
- (a) Decide if we need a separate Business layer.
- (b) Identify the responsibilities of the Business layer.
- (c) Do not mix different types of dat components in Business layer.

(f) Reuse common Business logic.

(g) Identify ~~for~~ the consumer of Business layer.

(h) Decide if we need a separate Business layer:-

- A separate Business layer helps to increase or improve maintainability of our application.

(i) Identify the responsibilities of the Business layer.

- Responsibilities includes processing business rules, transforming data, ~~data validation~~ etc.

- ① Do not mix other components in the Business logic =
→ Use the ~~Business~~ ^{Business layer} ~~DB~~ that decouples ^{Business logic from} the PL
and data access code.
- ② Reuse common business logic =
→ Use the Business layer that are reusable.
- ③ Identify the consumers in the business layer =
→ Determine how we will expose the business layer

WTF

④ Data Access layer :-

- ① → Data Layer Components
② Data Layer Approach
③ Data Layer Design Consideration

① Data layer Components :-

- DL components are :-
① Data access logic components
② Data helpers / Utilities
③ Service Agents.

② Data access logic components :-

- Data access logic components provides logic to access the database or data store.

③ Data Helpers / Utilities :-

- Data helpers / Utilities to assist in data manipulation, data transformation and data access within the layer.

① Service agent :-

- When we use external functionality in our application, then to communicate with it we need service service agents.

② Data layer Approach :-

- DL approach are :-

(a) Create an overall design Data layers.

(b) Design data access (logic) components.

(c) Design data helper components

(d) Design service agents components.

(a) Create an overall design Data layers.

→ Determine the Data access approach.

→ Determine how is to connect with data source.

(b) Design data access components :-

→ Identify the data source ~~you~~ we will access.

→ Decide on the method of access for each data source.

(c) Design data helper components :-

→ Identify the functionality

→ Consider custom Helper components for common problem.

(d) Design data service agents components :-

→ Use a appropriate tool to add a Service agents.

→ Determine how the service will be used in the application.

③ →

Data layer design consideration :-

→ DL design consideration are :-

①

Choose the data access technology.

②

Decide how to manage connections.

③

Consider security risk.

④

Consider performance and scalability objectives.

⑤

Choose the data access technology :-

→ Choose the appropriate data access technology depending on the type of data and manipulation of data in the application.

⑥

Decide how to

⑦

Choose the data access technology :-

→ Choosing the data access technology depends on

(i) what the type of data we deal.

(ii) how we want to manipulate the data in the application.

⑧

Decide how to manage connections.

→

We should decide how to create and manage all the connections to all data sources.

⑨

Consider security risk :-

→

Data access layer should protect the data from being attacked, stealing or corruption.

⑩

Consider performance scalability objectives :-

→

Scalability & performance objectives must be taken into account during design.

UV Imp

FF
→ MVC :-

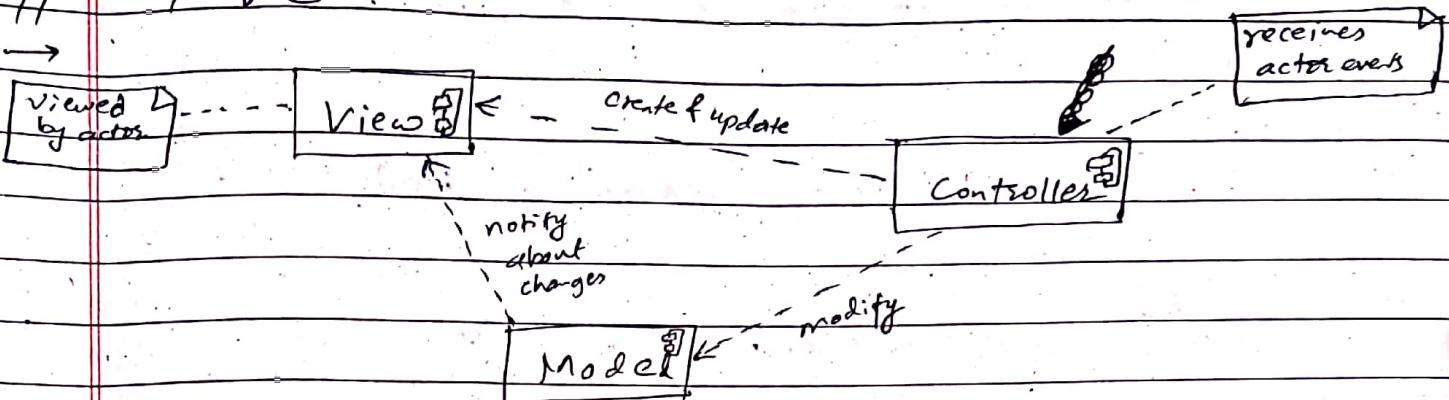


Fig = MVC (Model View Controller.)

→ MVC divides an interactive application into 3 parts :-

- (1) Model
- (2) View
- (3) Controller

(1) Model :-

- It contains the core functionality and data.
- It contains underlying classes whose instances are to be manipulated. Viewed and manipulated.

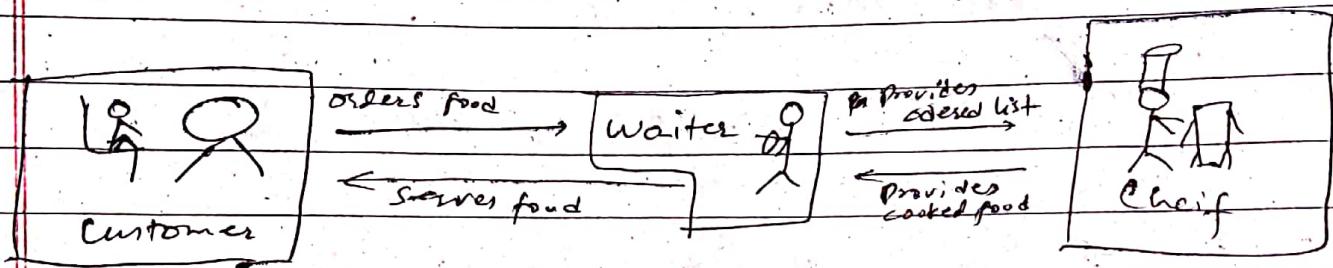
(2) View :-

- It displays information to the user.
- It contains the objects used to render the data from the model in the UI.

(3) Controller :-

- It handles the input from the user.
- It contains an object that handles user's interaction between View and model.

- It decouples the components and allows efficient code reuse.
- Web frameworks like Django and Rails use it.
- For example:
A restaurant with customer, waiter and chef



→ Benefits of MVC :-

- ① Organizes large scaled web-application.
- ② Supports Asynchronous Method Invocation (AMI)
- ③ They provide faster development process.
- ④ They are easy to modify.
- ⑤ Easy planning and maintenance.
- ⑥ SEO - Platform friendly platform
- ⑦ Supports Test-Driven development (T-DD)

→ Ex with Code :-

Model

```

int xPosition = 50;
int yPosition = 50;
String text = "PU";
  
```

View

PU

Controller

```

public void changeXPosition (int newXPosition)
public void changeYPosition (int newYPosition);
public void changeStringText
  
```

→ Code :-

Model

```
int xPosition = 50;
int yPosition = 50;
String text = "Example";
```

View

```
xPosition = 50
yPosition = 50
Text = Example
```

Controller

```
int public void changeXPosition (int newXPosition);
public void changeYPosition (int newYPosition);
public void changeText (String newText);
```

Java Server
Page

→ MVC architecture with the help of Servlet, JSP, PoJO classes & JDBC (Java database connectivity).

