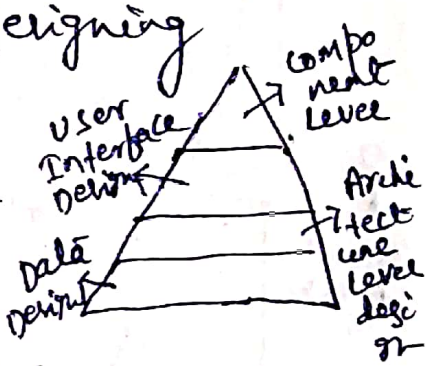
User Interface Design Model.

*) The design phase of s/w development is transforming the customer Requirements as described in SRS documents into design documents.

*) Designing a Model is a multiprocess that Represent the data structure, program structure, interface characteristics & procedurial details.

There are four types of designing Model.

1. — Data design Model
2. User Interface design Model
3. Architectural " "
4. Component level " "



User-Interface Design Model.

*) User Interface is the front-end application view to which user interacts with the s/w.

*) It determines how commands are given to the computer or program & how data is displayed on the screen.

*) The s/w becomes more popular if its user interface is:

- → Attractive.
- → simple to use.
- → Responsive in short time
- → clear to understand
- → consistent on all interfacing screens.

Types of User Interface.

Two types of User Interface

→ Text based User Interface or Command Line Interface
→ Graphical " " .

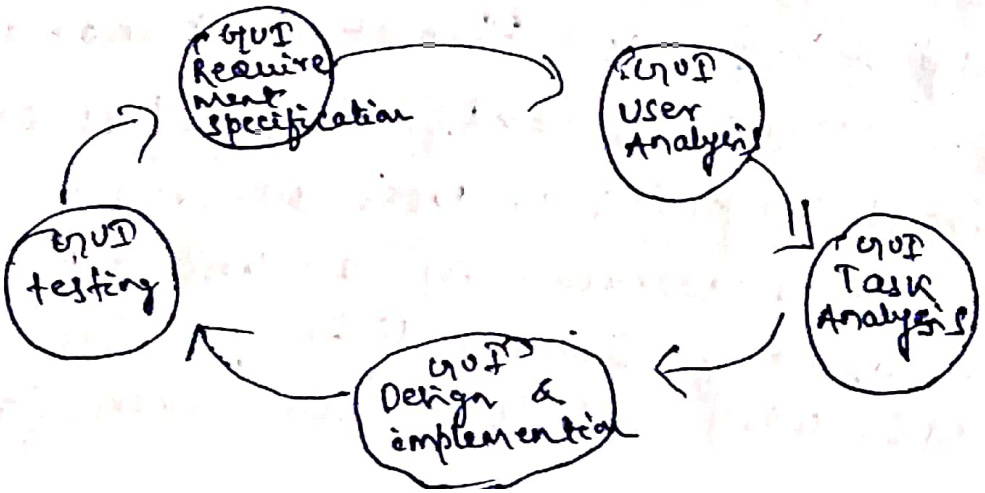
Text Based User Interface / Command Line Interface.

*) Primarily used keyboard handling data.
*) command line provide command prompt or coding tools where user types the commands towards the System.
*) User needs to remember the syntax of command and its use.
*) Used by technical people or programmer.

Graphical User Interface.

*) It provides the simple interactive interface to intract with the System.
*) It can be a combination of h/w & s/w
*) It is easy to learn as compared to command line interface
*) It provides multiple ~~user~~ windows to the user simultaniously to intract with the System.

User Interface Design Process / cycle.



① GUI Requirement gathering :- Designer Analyze all functionall & Non-functional Requirements mentioned & discussed with customer.

② User Analysis :- Designer Studies who is going to use the s/w GUI.

③ Task Analysis :- Designer analyze tasks, sub tasks & flow of the system.

④ GUI Design & Implementation :- Designer generate actual design by using different implementation tools like wavemaker, virtual Studio etc.

⑤ Testing :- Testing done by Inner designer team, Organization & different Stakeholders in project

User Interface principles.

*) User Familiarity :- based of user oriented terms & concepts.
e.g:- on windows OS, terms like Desktop, Document, Folder, Rename, power etc.

*) Consistency :- The system command & menus should have same format & parameter.
e.g. Group of fonts, Sizes, colors etc.

*) Minimal Surprise :- User never like to see the system working in an unexpected manner.
e.g. Group of similar type of features in MS word, File → New, Open, Save....

*) Recoverability :- User most often make mistakes while working with the system. ~~The~~ these mistakes can be Reduced
e.g. Undo, Rename.

*) User Diversity :- The interface should provide appropriate interaction facilities for the various types of the system user
e.g. Large font, privacy settings etc.

User Interface Design Golden Rules.

Three Rules are these

1. Place the user in control
   *) User should easily enter & exit
   *) ~~The~~ provides flexible interaction using keyboard, Mouse or touch screen etc.
   *) Display descriptive message & text.
   *) Allow users to customize the interface like short-cut keys etc.
   *) Hide technical internal details from user.
2. Reduce User's Memory load.
   *) Reduce demand on short term memory
   *) New features updated in system.
   *) The visual layout of the interface should be based on real world metaphore.
   *) Disclose ~~interface~~ information in a hierarchical & progressive fashion

3. Make the interface consistent.

*) Meaningful user tasks performed
*) Maintaining consistency among a family of applications.
*) Keep interaction Results the same as per the expectations

User Interface Design issues.

1. Response time :- Time between request & Response of the system. (should be very less)
2. Error Handling :- poor error msg may Result in rejecting the Syst. product.
3. Help facility :- User requires help when needs some information
4. Application Accessibility :- whether the applicn is simple to intract with or not.

Architectural Design Model.

*) It serves as a blueprint for a system.
*) The Architecture focuses on the early design decision.
*) Architectural Design consists of
   → Set of H/W & S/W Components that will perform a function required by the System. e.g. Database, Modules, Framework etc.

*) Set of Connectors :- helps in communication between the components.

*) Semantic Models :- Helps the designer to understand overall properties of the System.

Importance of S/w Architecture.

1. Security :- the System is secured against malicious users by Encryption or any other security measures due to layered S/w Architecture.
2. Performance :- Handles Request & Response of the page in minimum time.
3. Maintainability :- It uses Easily modifiable & Replaceable components
4. Safety :- Avoid critical functionalities in small component & improve Communication of the System.
5. Availability :- It includes Corresponding components, functionalities for handling the occurrence of any type of errors.

Decision of Architectural Design.

*) How can the System be distributed accross the network?
*) which approach can be used to structure the System?
*) which Architectural Style are suitable for the proposed System?
*) How can S/w Architecture be documented?
*) How the system be decomposed into modules!
*) How can Architectural design be analyzed!

Types of Architectural

1. Data centered Architecture:

~~*) Data is stored~~

*) Data stored at the center of this Architecture & is accessed by everyone.

*) Update, add, delete or the data present within the store

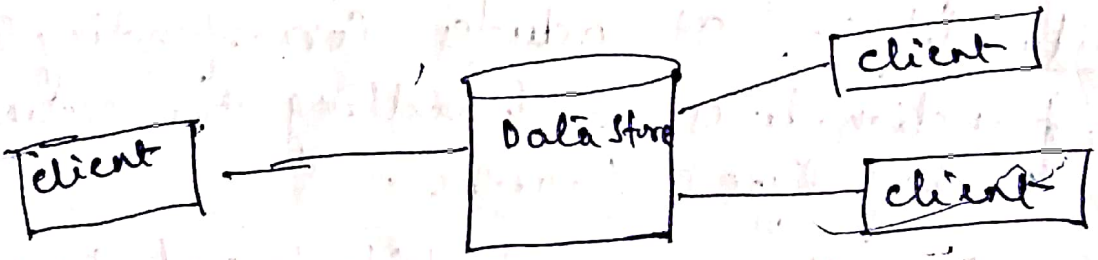
*) It is widly used in DBMS, library inform^n system etc.

Adv

1. Repository of data is independent of clients
2. It may be simple to add additional clients.
3. Modifi^n can be very easy.

Dis Adv.

1. Data Replication or duplication is possible.
2. changes in Data structure highly effect the clients.

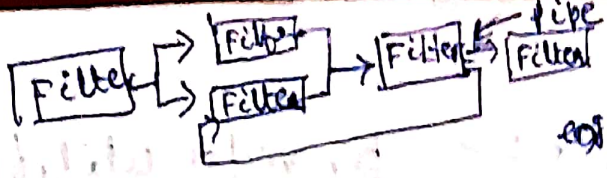


2. Data Flow Architecture.

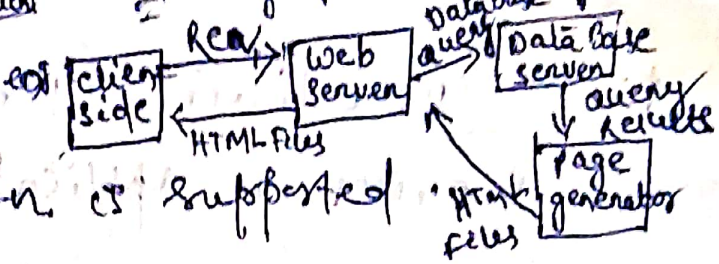
*) It is used when i/p data to be ~~transferred~~ transformed into o/p data through a series of computational manipulative components.

*) **Pipe** :- a connector which passes the data from one filter to next.

*) **Filter** :- Reads the data from its i/p pipes & performs its functions.



e.g: google search Engine.



Adv:
concurrent Execution is supported.

Disadv:
* Doesnot allow greater user engagement.

3. Object Oriented Architecture.

*) Objects are fundational building blocks for all kind of S/w applications.

Object:- object is an instance of class.
e.g. Student S,, Person.

class:- It defines all attributes, methods, which represents the functionality of the object

Encapsulation:- Binding similar types of elements

Abstraction:- Removal of irrelevant essentials from users.

Inheritance:- Deriving a new class from existing class.

polymorphism:- it has multiple forms.
e.g. draw graphic objects circle, Rectangle, triangle

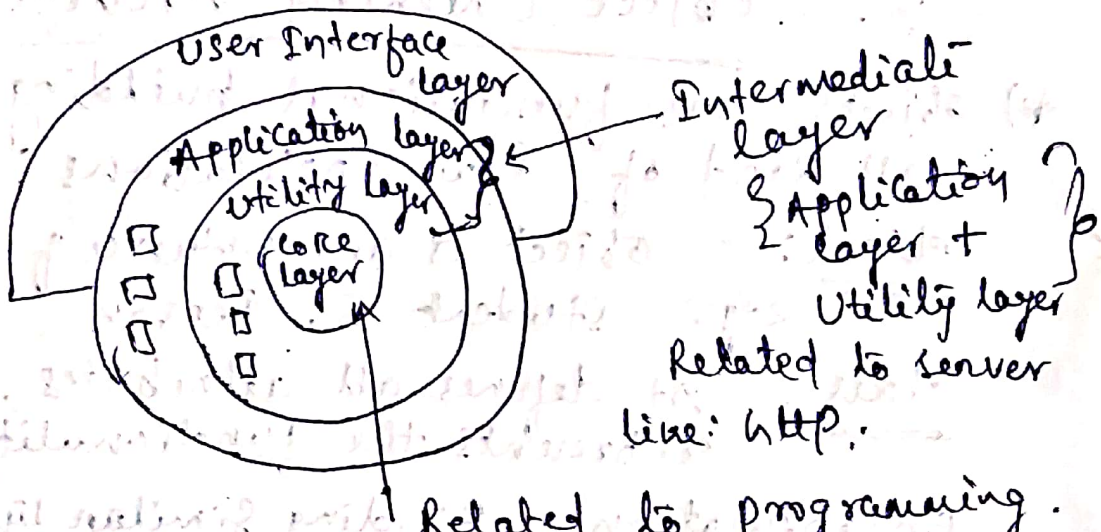
Message Passing:- Sending & Receiving data among objects through function parameter.

4. Layered Architecture.

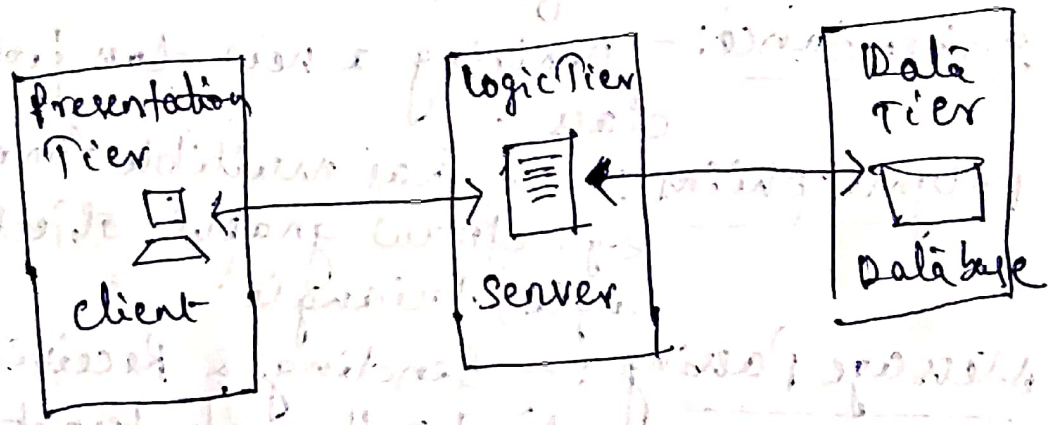
*) Data move from one level to another level
components manage
*) Outer layer:- User interface operations
*) Inner layer:- component will perform the O.S. interfacing.

*) Intermediate layer :- provides utility services & application s/w functions.

Ex :- E-Commerce Web Applications development like Amazon.



Ex :-



Component level Design.

*) A Component is a modular, portable, replaceable and reusable well-defined functionalities.
*) Component based Architecture ~~describes~~ focus ~~Concentrate~~ on breaking the S/w design into small individual modules.
*) It describes communication, interfaces, Algorithm & functionalities of each Component & regarding the whole S/w design.

Benifits of Component level Design.

*) It makes every module ~~Reusable~~ Reuseable
*) Reduces the cost as every module is "
*) It increases the Reliability of the System ~~as~~
*) It makes the System easy to maintain as we donot need to make changes in the entire System.
*) It describes each component's functionality more clearly.

characteristics of Component level Design

*) Modularity
*) Reusability - Component features are envisioned to be adaptable & Reusable to different projects.
*) Interoperability :- Components are connected through clearly defined interfaces, ensuring that S/w System maintain interaction continuity
*) Encapsulation :- Components have got encapsulate constructs
*) Scalability

Types of Components

*) **UI component**

→ User Interface components provide an easy way to encapsulate logic by combining visible elements such as buttons, forms & widgets.

*) **Service Components**

Service Component are the base of business logic or application services, in which they serve as the platform for activities such as data processing, authentication, & communication with external systems.

*) **Data Components**

Through data abstraction & provision of interfaces for data access, data components take care of data base interaction issues & provide data structures for querying, updating & saving data.

*) **Infrastructure Component**

The h/w element or Resource like Logging, caching, security & communication protocols which a s/w system depends on.

*) **Integration Component**

Integrated components for data communicate & data exchange between different modules

Component Design view.

1. Object Oriented view:
   - *) A set of collaborating classes in module
   - *) It view all the objects, methods, properties & functionalities mentioned in the classes.
   - *) It also view how classes in each module are connected in classes.

2. Conventional view
   - *) It is viewed functional element of a program that integrates the all logic & internal operation performed in a class.
   - *) It also view calling functions, parameter passing, data passing between modules.

3. Process Related view
   - *) It focus on reusability feature of the Component design.
   - *) Database & libraries are used to store pre-existed module.
   - *) This pre-existed data used for creating new module.

Component level ~~deg~~ design Guidelines

1. Component:- Naming convention should be established for components
   Ex:- calculator.
2. Interface:- Recognizes & discovers these independent ~~component~~ entitled as new components

*) interface provid important inform^n about communicatation & collaboration.

3) Dependencies & Inheritance :-

*) Models any dipendencies from left to right and inheritance from top to buttom.
*) use complete concept of cohesion & coupling

Component level Design for web Apps.

*) What is web App component?
*) A well defined cohesive function that provides data processing for an end users.
*) A cohesive package of control and functionality that provide the enduser with some required capability.
*) Web Applications include online forms, shopping carts, word processors, spreadsheets, video, & photo editing, file conversions etc.

Component Design for Web App.

1. Content design at Component level.

*) It focus on objects & their collaboration in packaged for presentation to a web App end user.
*) It should be ~~turned~~ tuned to the characterstics of the ~~webApp~~ Web App to built.
*) It may be necessary to organize content in a way that allows easier reference & design.

2) Function Design at component level.

*) Perform localized processing to generate content & navigation capability in a dynamic fashion
*) Provide sophisticated database query & access.
*) Establish data interfaces with external corporate systems.

3) Deployment Design at component level.

*) It indicates how s/w functionality & subsystem will be allocated within the physical computing environment that will support the s/w
e.g. personal computer, internet connectivity, server, control panel like security etc.

Questions

1. ~~Estab~~ Enlist the golden Rules for user interface design.
2. Explain User interface design principles
3. Explain the user " " issues.
4. Explain Data flow Architecture Style.
5. Explain layred Architecture Style.
6. Explain in~~detail~~ data centered Architecture
7. What is Architecture? Explain Architecture context diagram

(8) Explain object Oriented view of component level design with suitable Example.

(9) Explain guidelines of component level design.

(10) Explain component level design for Web Apps