18/12/23

ASSIGNMENT 3

PAGE NO.: 73

Q.1 Describe component level software design with its significance.

-> Component-level software design is a phase in the software development process that involves breaking down the system's high-level design into smaller, more manageable units called components. A component can be thought of as a modular and self-contained unit of software that performs a specific functions. These components are designed to interact with each other to form the complete software system.

- Significance :-

i) Modularity : It promotes modularity by breaking down the software system into smaller, independent components.

ii) Reusability : Components are designed to be reusable across different parts of the software system or in different projects which reduces redundancy in code.

iii) Maintainability : Since each component is designed to perform a specific function, maintenance becomes easier because changes or updates can be ~~done~~ made to a specific component without affecting the whole system.

iv) Scalability: They are often more scalable. If additional functionality is required, new components can be added without affecting existing ones.

v) Collaboration: It facilitates parallel development and collaboration among team members. Different teams can work on different components simultaneously.

Q.2 State and define golden rules for interface design.

→ There are three golden rules for interface design are fundamental principles aimed at creating user-friendly and efficient user interfaces.

i) Place the user in Control.

Users should feel that they are in control of the system and can easily navigate through it. Interface design should provide clear and understandable options, actions and feedback to users. Users should be able to initiate actions and make decisions confidently.

ii) Reduce the User's Memory Load.

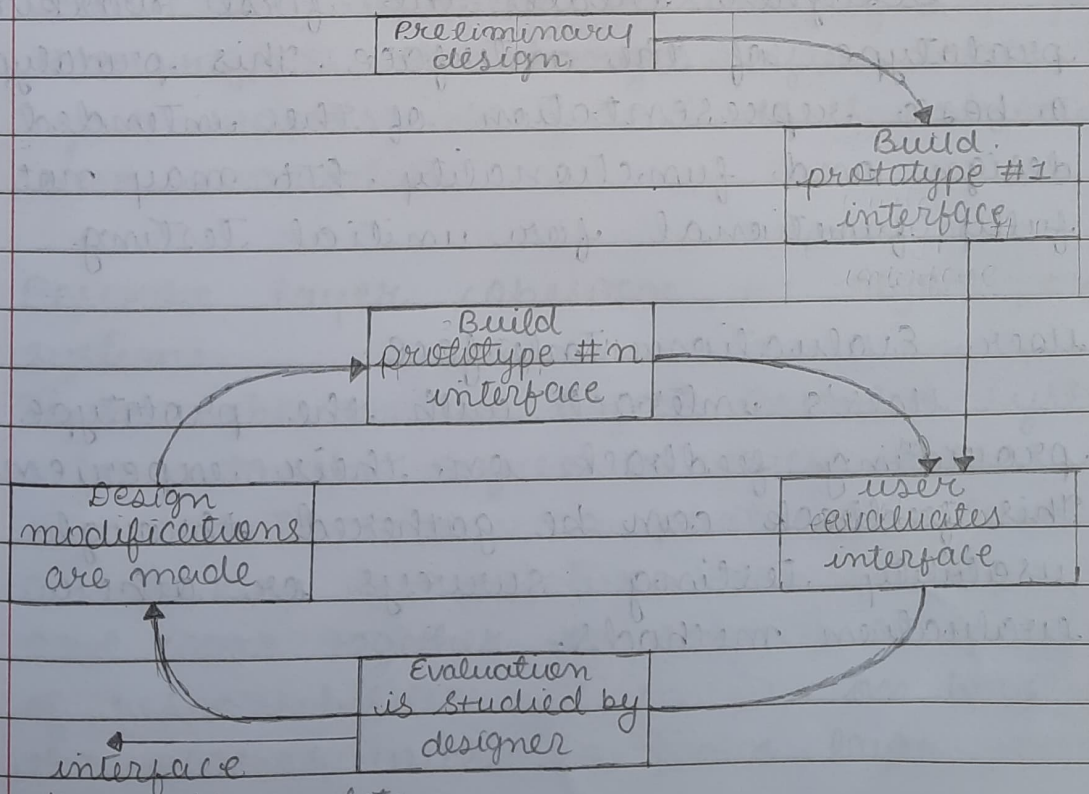
Design interfaces in a way that minimizes the amount of information

users need to remember. Users should not be burdened with excessive memorization of system features. Instead, provide cues, reminders and easily accessible information so that users can perform tasks without relying heavily on their memory.

iii) Make the Interface Consistent

Maintain consistency in design elements, terminology and interactions throughout the interface which helps users to build a mental model of the system, making it easier ~~to~~ for them to predict how the interface will behave.

Q.3 — 4 Explain interface design evaluation cycle.



The Interface Design Evolution cycle is a structured iterative process that involves several key steps to develop and refine a user interface. This cycle is a continuous loop, allowing designers to gather feedback, make improvements and iterate on the design.

i) Preliminary Design.

This phase involves conceptualizing and planning the initial design of the interface. Designers outline the basic structure, layout and functionality based on user requirements and business goals.

ii) Build Prototype #1 interface.

Desogner create the first working prototype of the interface. This prototype is a basic representation of the intended design and functionality. It may not be fully functional for initial Testing.

iii) User Evaluation Interface.

User's interact with the prototype, providing feedback on their experience. This feedback can be gathered through usability testing, surveys or other evaluation methods.

iv) Evaluation is studied by Designer.
Designers analyze the feedback received from users during the evaluation phase. They identify strengths, weaknesses and areas for improvement in the interface

v) Design Modifications are Made.
Based on the insights gained from user evaluations, designers make adjustments to the interface which involves refining the layout, enhancing functionality, addressing usability issues or incorporating additional features.

vi) Build Prototype #n interface.
After making modifications, designers create an updated version of the prototype that reflects the changes. After that the cycle repeats itself and it is subjected to user evaluation

Q.4 Describe layer cohesion in object-oriented systems.

→ Layer Cohesion in Object-Oriented systems refers to the degree to which classes and components within a particular layer of a software architecture are closely related and work together to achieve a specific set of responsibilities. It focuses on how well the classes within a given layer collaborate

and share common goals and functionality. There are generally three types of layer cohesion:

i) Functional Cohesion
Classes within a layer collaborate to perform a specific set of related functions or operations.

ii) Sequential Cohesion
Classes within a layer collaborate in a sequential manner/order, where the output of one class becomes the input for the next class.

iii) Communicational Cohesion
Classes within a layer collaborate by sharing a common set of data or information.

Q.5 State and describe any two views of software components.

→ Different views are used to understand and document various aspects of a software system. There are basically three views:

i) Traditional View

ii) Object-Oriented View.
It emphasizes the organization of

software components based on the principles of Object-Oriented Programming. In this view, the software is modeled as a collection of interacting objects, each encapsulating data and behaviour. Objects communicate with each other through well-defined interfaces and the relationship between objects are defined by concepts such as inheritance, polymorphism & encapsulation.

iii) Process-Related View.

The Process-related view is concerned with how software components interact during runtime. It focuses on the dynamic aspects of the system, including concurrency, communication and coordination of processes. This view is particularly important in systems where multiple tasks needs to work together to achieve a common goal. ~~If~~ It often involve concepts like inter-process communication, synchronization and concurrent execution.

Q.6 Discuss various layout issues in design of web applications.

→ Layout issues in the design of web applications play a crucial role in determining the visual ~~hierarchy~~ hierarchy, user experience and overall usability of a website.

i) Responsive Design.

Inconsistency in layout across different devices and screen sizes. Implement responsive design to ensure that the layout adapts seamlessly to various screen sizes like desktops, tablets & smartphones.

ii) Whitespace and Clutter

Too much or too little whitespace can impact readability and visual aesthetics.

iii) Navigation Design

Poorly designed navigation can confuse users and hinder their ability to find information.

iv) Loading Times.

Slow-loading pages can frustrate users and impact their experience negatively.

v) Consistency.

Inconsistent layouts across pages or sections can create a disjointed user experience.

vi) Grid layout and Alignment

Inconsistent alignment and lack of a grid structure can lead to a visually

disorganized layout

vii) Hierarchy and Visual Flow.

Lack of a clear visual ~~flow~~ hierarchy can lead to confusion about the importance of different elements on a page.

Q.7 Discuss pattern languages and repositories.

→ Pattern languages and repositories are valuable tools for capturing and sharing design knowledge, promoting best practices and fostering collaboration among designers and developers.

→ Pattern Languages:

A Pattern Language is a structured and organized collection of design patterns that provide solutions to common design problems. Each pattern in a pattern language describes a recurring problem, its context and a proven solution.

Pattern

Patterns are modular and can be combined to address complex design challenges. Patterns are interconnected and may reference or build upon each other. They capture high-level design concepts and principles, promoting abstraction for broader applicability.

→ Repositories

A repository is a centralized and organized collection or database that stores design patterns, guidelines and related artifacts. It serves as a knowledge hub where designers and developers can access, contribute to and retrieve design-related information.

Repositories allow users to search for specific patterns based on keywords, categories or tags. Users can contribute new patterns, provide feedback and collaborate with others within the repository.

Q.8 Explain design focus and design granularity in web application design patterns.

→ Design Focus

It refers to the primary concern or aspect of the application that a particular pattern addresses. It is the central theme or purpose of a ~~different~~ design pattern. It also helps in identifying which patterns are most suitable for solving particular problems. Different levels of design focus:

i) Information Architecture Patterns: These patterns relate to the overall structure of the information space in a web application and how users interact with the information.

ii) Navigation Patterns: These patterns define the link structures for navigation, such as hierarchies, rings, tours, etc.

iii) Interaction Patterns: These patterns contribute to the design of the user interface, addressing issues, describing link destinations and providing feedback on ongoing interactions

iv) Presentation Patterns: These patterns assist in presenting content to users through the interface, focusing on organizing user interface control functions and establishing effective content hierarchies.

v) Functional Patterns: These patterns define workflows, behaviours, processing, communication and other algorithmic elements.

→ Design Granularity:

It refers to the level of detail at which a design pattern operates. It indicates how broad or specific the scope of a pattern is in addressing design issues.

Different levels of granularity:

i) Architectural patterns: These patterns deal with the overall pc structure, specifying relationships among different components and defining rules for their interactions.

ii) Design Patterns: These patterns addresses ~~specific~~ specific elements of the design for component-to-component communication.

iii) Component Patterns: These patterns relate to individual, small scale elements of a web application, including interaction, navigation ~~and~~ or functional elements.

Q.9 Explain navigation semantics with an example of web application.

→ Navigation semantics refer to the principles and practices governing how users move through and interact with the application's interface. Effective navigation semantics contribute to usability, user satisfaction and successful task completion.

i) Navigation Elements.

They are the components that allows users to move between different sections, pages or features.

eg: Consider an online store. Navigation elements may include a top navigation bar with links to product categories (eg. Home, Clothing, Electronics, Kitchen)

ii) Search Functionality

It allows users to find specific content quickly, providing an alternative

means of navigation.

eg: In the Online store, users can use the search bar to find products directly by entering keywords, product names (eg. shoes).

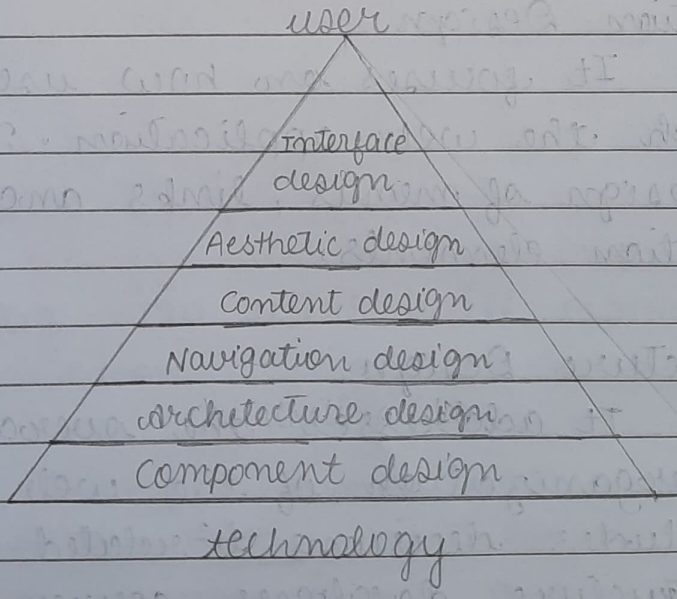
iii) User Feedback.

Providing feedback, such as visual cues or notifications, helps users understand the outcome of their navigation actions.

eg: After adding an item to the cart, a brief notification appears confirming the action and the cart icon may visually indicate the updated item count.

Q.10 Explain design pyramid for web applications.

→ The design pyramid suggests a hierarchy of design considerations, ~~emphas~~ emphasizing the foundational aspects that support the overall user experience.



i) Interface Design (Top Layer)

It focuses on how users interact with the web application. This includes the layout of the user interface, user interaction and the overall ~~des~~ user experience.

ii) Aesthetic Design

It deals with visual appeals of the web application. It includes considerations such as colors, schemes, typography, imagery and other design elements that contribute to the overall ~~st~~ aesthetics.

iii) Content Design

Content design forms the foundation for the layers above. In ~~above~~ involves creating and organizing the textual, visual and multimedia content presented.

iv) Navigation Design

It focuses on how users move through the web application. This includes the design of menus, links and other navigation elements.

v) Architecture Design

It addresses the overall structure and organization of the web application. It includes decisions ~~to~~ related to the backend infrastructure, databases, server-side logic.

vi) Component Design (Base Layer)

It focuses on the design of individual elements or modules within the web application which includes button, forms, cards.

Q.11 Discuss Object Oriented Hypermedia Design Method (OOHDM).

→ The Object Oriented Hypermedia Design Method (OOHDM) is a design methodology specifically developed for hypermedia systems, where navigation and linking are essential components. It was developed in the 1990s by Schwabe and Rossi.

- Methodology Phases:

i) Conceptual Phase.

Identify the nodes and links that represent content and navigation within the application.

ii) Navigational Design Phase.

Specify the navigational structure, determing how users will navigate through the application.

iii) Abstract Interface Design Phase.

Defines the appearance and behaviours of navigation spaces.

iv) Concrete Interface Design Phase
Address the visual and interactive aspects of the user interface.

v) Implementation Phase
Implement the hypermedia application based on the design specifications developed in earlier phases.

Q.12 Explain MVC architecture.

→ MVC (Model-View-Controller) is a software architectural pattern widely used in the development of user interfaces, particularly in web applications. The pattern divides the application into three interconnected components:

i) Model
It represents the application's data and business logic. It encapsulates the data structure, performs data processing and manages the application's state and behaviour.
- Manages and maintains the application's data
- Implements business logic, rules & algorithms
- Notifies observers of changes in the data

ii) View
It is responsible for presenting the data to the user and capturing user input

- Displays information to the user based on the data from the model.
- Captures user input.
- Sends user actions to the controller for further processing.

iii) Controller.

It acts as an intermediary between the Model and the View. It receives user input from the view, process it and updates the Model accordingly. The Controller also updates the view based on the changes in the Model.