SE:- Assignment - 2

Date
Page

Unit - 3 (User Interface Design)

(1) Role of UI :

⇒ User interface plays a crucial role in the success of software projects.

→ User interface is not only the reflection of the end users.

→ If the portal / website is intended for particulars demographics then we need to choose the colors appriciated by that region.

Additionally each color has its own meaning that needs to be considered while choosing the color.

- Black : mystery, secrecy
- Blue : coolness, peace
- Brown : strength, stability
- Gray : maturity, realiability
- Green : life, naturalness, health
- Orange : warmth, courage
- Red : danger, energy, power, aggression.
- White : Purity, freshness, peace
- Yellow : happiness, brightness, joy.

- UI allows user to interact with system and manipulate the system (machine)

(2) Three Basic Components of UI :

⇒ 1) End Users
2) User Interface
3) System (machine)

(User Interface)
→ A UI helps the people (Users) to interact with a machine (System).

→ User Interface includes its own software and hardware components which helps the users to interact with the machine (System)

→ The goal of UI is to effectively control the machine (System) at the user's end and pass the feedback from the machine (System) back to the user in efficient manner which helps User to make decisions

→ This means the user needs to provide minimal input to achive the desired output and stopping machine to provide undesired output.

User	⟷	User Interface	⟷	System
- Human		- Hardware - Software		- Machine - Software

→ User interface is also called as human-computer interface (HCI) and man-machine interface (MMI)

Elements of User Interface :

1.) Users
2) Tasks
3) Contents
4) Environment

1) ~~Users~~ :

Users interact with the system through the user interface.

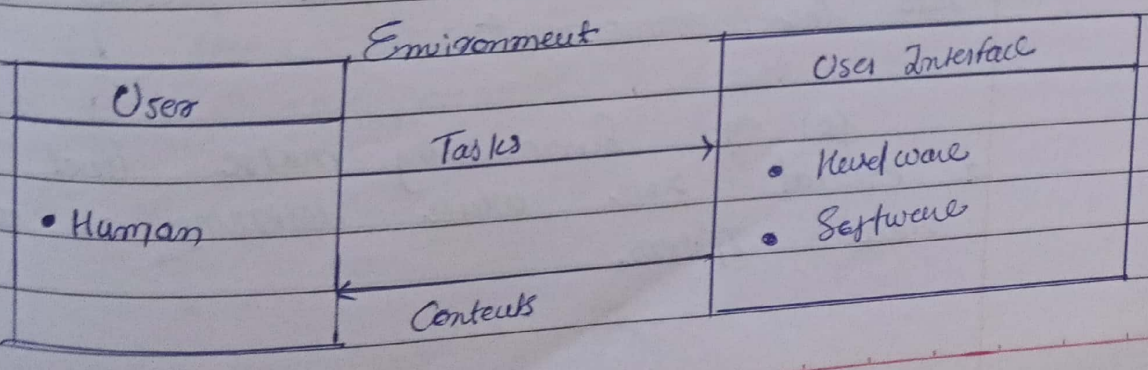
2) ~~Tasks~~ :

Tasks are the activities performed by the users to interact with the system. Contents are the output of the system that is displayed to the users through the user ~~face~~ interface.

The Environments in which these tasks are executed are also part of the user interface.

⇒ Analysis based on these elements :

1. User Analysis
2. Task Analysis
3. Contents Analysis
4. Environment Analysis



1) User Analysis:

User Analysis helps to understand the end users as well as how the end users use the system which will help them to design the UI in a better way.

2) Task Analysis:

Tasks Analysis helps us to understand the tasks done by the end users under a specific circumstance.

It helps to understand the overall flow of the users as well as the tasks performed by the users.

3) Content Analysis:

Content analysis is an important element of the overall user interface as it directly satisfies the end users.

4) Work Environment Analysis:

Work Environment analysis plays a crucial role as the designed product is fitting into the environment.

for eg. Surrounding noise level plays a crucial role while designing an audio/video system.

Q) Explain Three Golden Rules of User Interface Design.

→ (1) Place Users in Control
(2) Reduce Users' memory load
(3) Make the Interface Consistent

1) Place Users in Control:

It includes providing customizable screen, based on the user profile, proper help + navigation, interactive display system.

By providing the control to the users, they feel as if they own a system and will start using it.

2) Reduce Users' Memory Load

Sub rules:

→ Rely on Recognition, not recall (recognition only).
→ Provide Interface frequency (shortcut).
→ Provide Visual Clarity (organize).
→ Promote an object-action syntax (intuitive)
→ Use real-world Metaphors (transfer)
↳ Metaphors are the words used to represent an action as word which is not related to its own literaly meaning

(forgiving)

→ Provides Redo, Undo and Defaults
→ Users progressive disclosure (context)
↳ display only what is basics needs.

3) Make the Interface Consistance:

Sub Rules:

→ Provide aesthetic appeal and integrity (appliance)
↳ feel something by looking into a something

→ Encourage Exploration (predictable)

→ Sustain the context of users' tasks (continuity).

→ Maintain consistency within and across products (experience)
↳ same experience in all section (products) like background color, fonts etc.

→ Keep Interaction results the same (expectations)

Common User Interface Design Issues:

1) Improper error messages to the users.
2) No ~~Menu~~ proper Menu Labeling.
3) Improper help facilities to the users.
4) Response time of the system.

1) Improper Error Messages to the Users:

→ ~~Easily~~ Mesa ~~Should~~ Message should easily understandable.
→ It should not blame user
→ It should display visual ~~cue~~ cue (clue)
→ display reason of the error.
→ Indicate the advice to overcome the situation # Indicate (display)
→ Also Indicate the successfully done transaction in form of message.

2) No Proper Menu Labeling:

Menu Labeling should address the following:

→ All menu labels are self explanatory
→ Shortcut options available for menu actions (eg: ctrl+o → Open)
→ Can Menus be customized by the users
→ Sub Menus are ~~consti~~ consistent with the main menu.

3) Improper help facilities to the Users:

⇒ Help manuals should describe all the scenarios of the system and it should be written in easy language so that any novice users can understand and use the system.

4) Response time of the System.

→ Response time of the system is very important and it needs to be taken care even while designing the User Interface.

Unit - 4 (Software Coding & Software Testing)

(1) Software programming principles.

→ 1) Validity :

The program must given the current correct result which is valid.

eg: 4+5 = 8 answer must be 9.

2) Consistency :

The program must do repeatedly what it intends to do

eg : (2+5+1) = 8 then (1+5+2) also must be 8 in output of program.

3) Maintainability :

The program must be easily changeable (modification) and should have proper documentation

4) Readability :

The program must be easily readable so that it is easily maintainable

5) Usability :

The program must be usable for the specific purpose without any trouble.

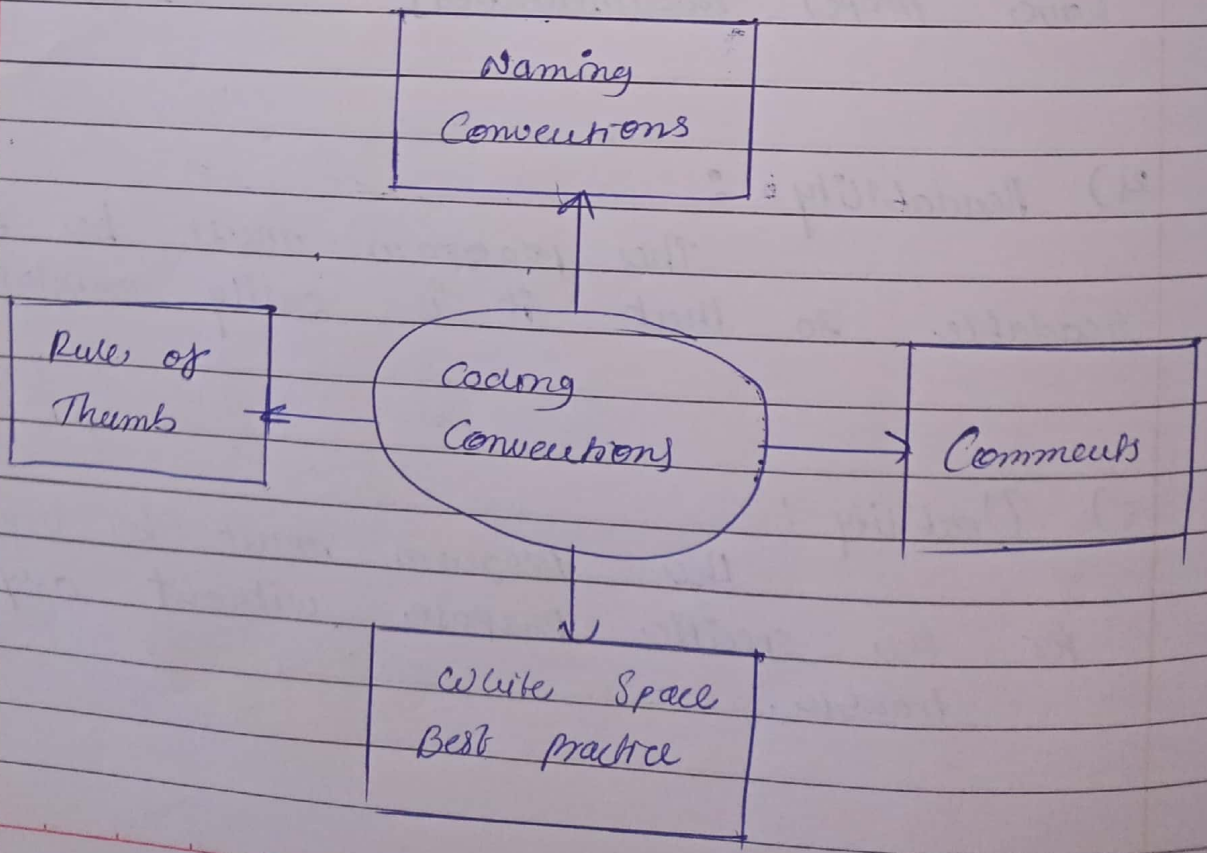
(2) Coding Conventions:

Coding Conventions are a set of best practices that helps to write the software code in an efficient manner.

1) Naming Conventions:

Set of rules for choosing the characters sequence to be used for identifiers (names)

- → Filename,
- → Foldername,
- → Variable name
- → Function (Method) Name
- → Length of the identifiers
- → Letter case and numerals
- → Multiple word identifiers
- → Hybrid conventions.



2) Rules of Thumb & Programming Principles

→ Choose Appropriate data type and size
→ Avoid declaring all variable as global
→ keep specific name
→ Avoid multiple-purpose variable and methods (use for one and only purpose)
→ Avoid writing 'public class' for security purpose.
→ Create Connection object late as posible and release it early as possible.
→ Avoid using database connection using specific user's credential. we cannot reuse those connections.

3) White Space & Best Practice:
These concept improve the readability of code.

⇒ Blank Lines:

i) Two blank lines can be used between sections
ii) Two blank lines can be used between class definitions within the source code
iii) Two blank lines can be used between Interface definitions within the source code
iv) One blank line between method before a block within the source code.

⇒ Blank Spaces:
→ A blank space should always be after comma in any argument list. keyword followed by brackets must be separated by a space. Binary operators should be separated by blank spaces for readability.

4) Comments:

→ Properly commented code serves no purpose for the compilation as well as executing the code, but it improves the readability of the code.

→ It also helps to understand the purpose and business logic behind the code.

(3) Explain the Guidelines for comments in Software coding.

⇒
- Keep the comment always up to date.
- write the comments first before chewing the actual code
- write Commenteds at the beginning of every method, indicating the purpose of the method, Assumptions and limitations.
- Try to Avoid End-of-Line Comments.
- Before deployment, remove all unnecessary and temporary comments.
- write comments for what the variable is gonna do (purpose of variable)
- Avoid writing series of asterisks in the comments. Next

- Use complete sentences.
- Comments should add to help to understand the code, not to confuse.

- Use uniform style throughout the code for comments. (punctuation & structure)

(4) Software Testing and Software Testing objectives.

→ Software Testing:

It is a process of evaluating and verifying that a software product or application does what it is supposed to do.

→ Objectives:

Main or first objective of software testing is to prevent bugs from getting into the system. It's commonly called verification testing.

→ The other obvious objective is to find and fix bugs.

↳ whether the software products meets the specification

↳ whether the software product is fit for the purpose it was designed.

↳ whether there are faults which would cause failure during the operations of the software under various loads, user inputs, and conditions.

(5) Verification Testing & Validating Testing.
Issues Impacting in Verification & Validation Testing.

→ Ⓐ Verification Testing :
It is a type of testing that prevent bugs from getting into the system.

→ Verification happens Intenti Internally.

→ Impacting Issues of Verification Testing :

- Lack of proper documentation,
- Lack of understanding requirements,
- Not following process

⇒ Validation Testing :

The testing activities that are done to evaluate the software in the executable mode can be called as validation testing.

⇒ Impacting Issues of Validation testing :

- Lack of time
  (Tester must ensure to get enough time to test)
- Software Crashes ;

4 Missing Critical Defects :

→ Validating Testing basics phases :

- Phases Unit Testing
  (Usually done by developers)
- Integration Testing
  (Ensure the interactions with the software modules.)

- Functional testing :
  to verify end-to-end functionalities of the modules.

- System Testing :
  All system - software, hardware, external interfaces tested as per the specifications.

- Acceptance Testing :
  (Done by Customer (Client).
  Accept the product after execute all the test cases. for evaluating the software product.