Robert A Conchada
DICT-3 : Software Engineering

Lesson 10

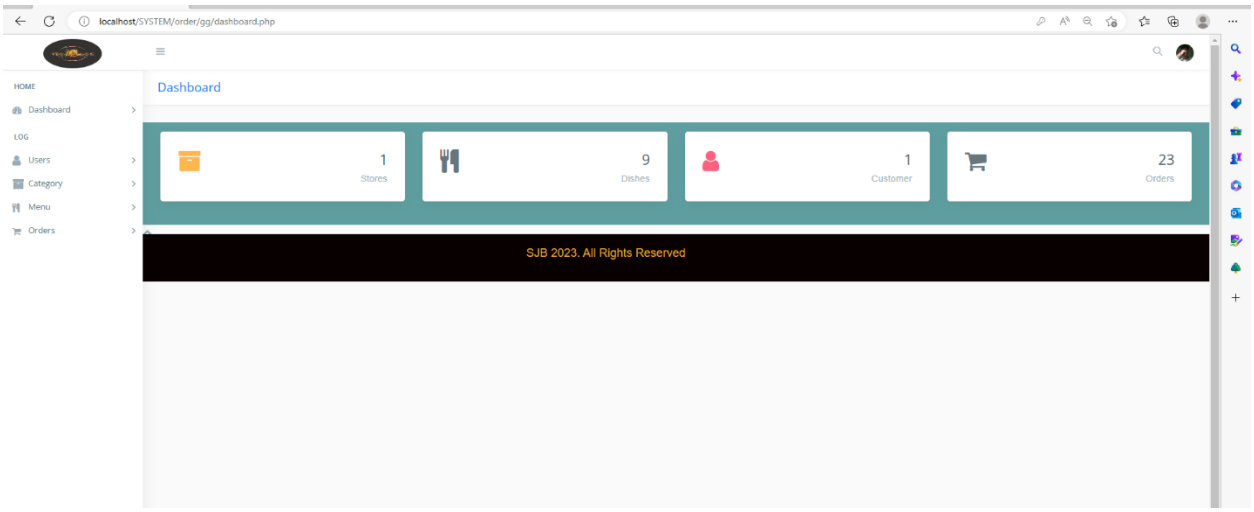
Activities/Assessment

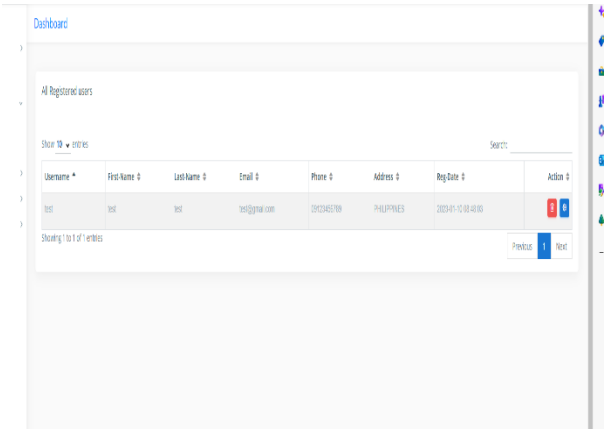
1. **Define what user interface is.**

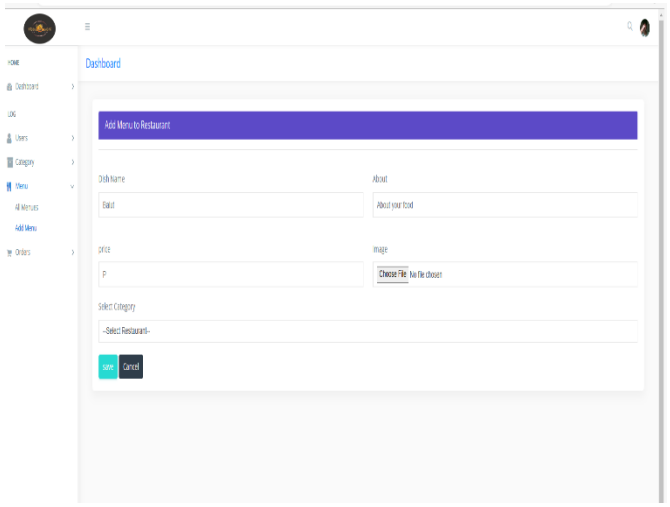
User Interface (UI) refers to the visual and interactive elements of a software application or website that allow users to interact with and control the system. It includes graphical components such as menus, buttons, text boxes, and icons, as well as the layout and design of the overall interface. The primary goal of a user interface is to make it easy and intuitive for users to navigate and accomplish tasks within the system.

User interface is part of software and is designed in such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction. UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:
- Attractive
- Simple to use
- Responsive in short time
- Clear to understand
- Consistent on all interfacing screens UI is broadly divided into two categories:
- Command Line Interface
- Graphical User Interface







2. Discuss the User Interface Golden rules.

The User Interface Golden Rules are a set of guidelines for designing effective and user-friendly interfaces. They are as follows:

- Strive for consistency: Ensure that the interface behaves consistently across different tasks and contexts. Consistency in the layout, terminology, and user interactions promotes ease of use and reduces the learning curve for users.

- Enable frequent users to use shortcuts: Experienced users often want to complete tasks quickly, so it's important to provide keyboard shortcuts, hotkeys, and other features that help them navigate and perform actions more efficiently.

- Provide informative feedback: The system should provide feedback to users about their actions and the current status of the application. This helps users understand what is happening, whether their actions were successful, and what steps they need to take next.

- Design dialogues to yield closure: Dialogues, such as confirmation messages or error messages, should be designed to provide a sense of closure. They should clearly state the result of the action and what the user can do next.

- Prevent errors: Good interface design anticipates and prevents errors. This includes designing error messages that are clear and easy to understand, as well as validating user input to prevent mistakes.

- Permit easy reversal of actions: Users should be able to easily undo actions or navigate back to previous steps. This reduces frustration and the risk of users making mistakes.

- Support internal locus of control: Users should feel in control of the system and its functions. This means providing clear and easy-to-understand instructions, as well as designing the interface in a way that enables users to complete tasks at their own pace.

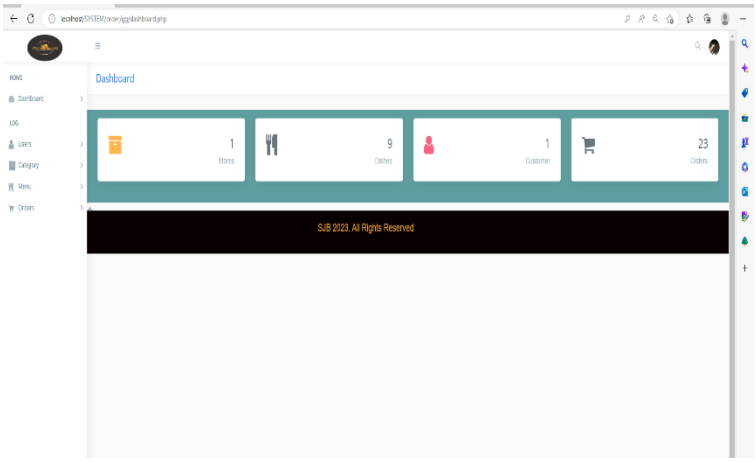
- Reduce short-term memory load: Interfaces should be designed in a way that minimizes the amount of information users need to remember at any given time. This includes using clear and concise language, providing visual aids, and breaking complex tasks into smaller steps.

By following these guidelines, designers can create interfaces that are intuitive, efficient, and enjoyable for users.

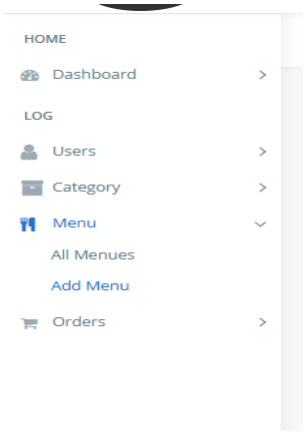
3. **With your group, re-create the mock-up design with the user interface golden rules.**

- Strive for consistency:

Ensure that the interface behaves consistently across different tasks and contexts. Consistency in the layout, terminology, and user interactions promotes ease of use and reduces the learning curve for users.



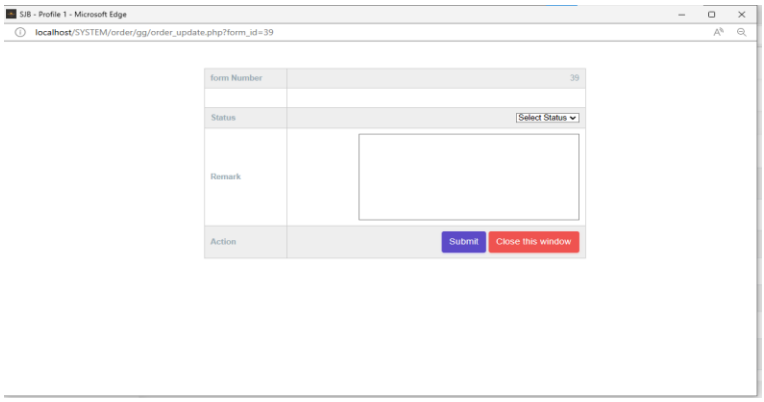
- Enable frequent users to use shortcuts:



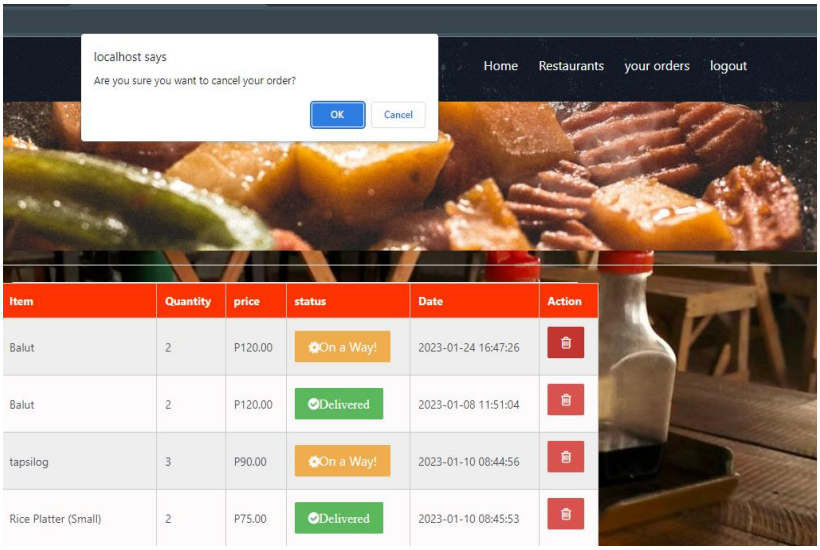
Experienced users often want to complete tasks quickly, so it's important to provide keyboard shortcuts, hotkeys, and other features that help them navigate and perform actions more efficiently.

- Provide informative feedback:

The system should provide feedback to users about their actions and the current status of the application. This helps users understand what is happening, whether their actions were successful, and what steps they need to take next.



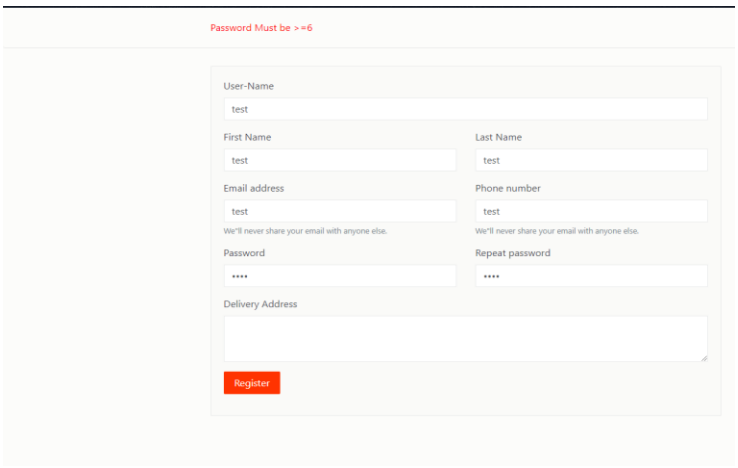
- Design dialogues to yield closure:



Dialogues, such as confirmation messages or error messages, should be designed to provide a sense of closure. They should clearly state the result of the action and what the user can do next.

- Prevent errors:

Good interface design anticipates and prevents errors. This includes designing error messages that are clear and easy to understand, as well as validating user input to prevent mistakes.



- Permit easy reversal of actions: Users should be able to easily undo actions or navigate back to previous steps. This reduces frustration and the risk of users making mistakes.

- Support internal locus of control: Users should feel in control of the system and its functions. This means providing clear and easy-to-understand instructions, as well as designing the interface in a way that enables users to complete tasks at their own pace.

- Reduce short-term memory load: Interfaces should be designed in a way that minimizes the amount of information users need to remember at any given time. This includes using clear and concise language, providing visual aids, and breaking complex tasks into smaller steps.