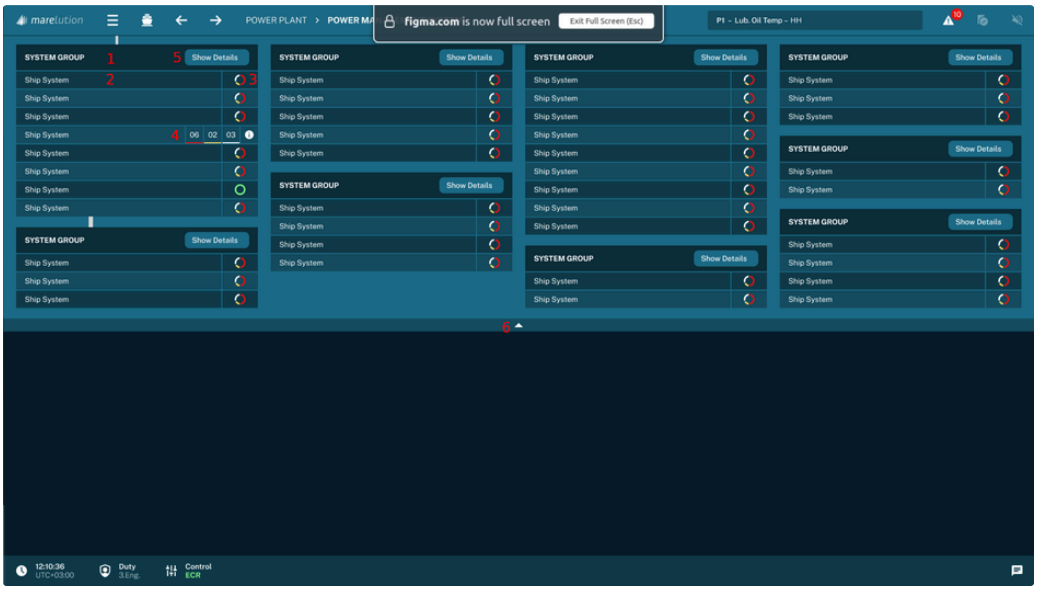
Requirements Ship Systems Dropdown Menu



Requirement	Type
The ship system drop down menu is the main point of navigation that provides navigation options and also provides information about the respective page.	Info
The actual header and footer implementation is not scope of this work. The task is considered completed when a modularly programmed dropdown menu that can be integrated elsewhere is shown to be functional with a dummy header and footer (e.g. rectangles).	Info

The dropdown menu must be generated from a json file that defines the different columns, the main groups (1) and the sub groups (2) as exemplary defined here:
 	
`{`
`  "columns": [`
`  {`
`  "index": 1,`
`  "systems": [`
`  {`
`  "system_group": "System A",`
`  "ship_systems": [`
`  "Ship System A1",`
`  "Ship System A2",`
`  "Ship System A3"`
	Must

```
          ]
        },
        {
          "system_group": "System B",
          "ship_systems": [
            "Ship System B1",
            "Ship System B2"
          ]
        }
      ]
    },
    {
      "index": 2,
      "systems": [
        {
          "system_group": "System C",
          "ship_systems": [
            "Ship System C1",
            "Ship System C2",
            "Ship System C3"
          ]
        }
      ]
    },
    {
      "index": 3,
      "systems": [
        {
          "system_group": "System D",
          "ship_systems": [
            "Ship System D1",
            "Ship System D2"
          ]
        },
        {
          "system_group": "System E",
          "ship_systems": [
            "Ship System E1",
            "Ship System E2",
            "Ship System E3"
          ]
        }
      ]
    },
    {
      "index": 4,
      "systems": [
        {
          "system_group": "System F",
          "ship_systems": [
            "Ship System F1",
            "Ship System F2",
            "Ship System F3"
          ]
        }
      ]
    }
  ]
}

```

The dropdown menu must smoothly slide down from behind the headerbar. Such an animation is considered smooth: Simple Minimal Dropdown Animation	Must
Every ship system has a number of faded alarms, suppressed alarms, invalid states and active alarms.	Info
The circle on the right of the ship system name (3) must represent these states the following way: • When faded, suppressed, invalid and active counts are zero, the circle must be green. • If at least one of the counts is non-zero, the colors in the circle must represent the respective counts where red represents the active count, yellow represents the invalid count and white represents the combined count of faded and suppressed.	Must
When *Show Details* is pressed, the respective actual counts must be displayed for all ship systems of the system group (4) instead of the circle.	Must
Pressing the info button (4) must trigger a function *display_ship_system_info* with the ship system name as argument.	Must
Implementing the *display_ship_system_info* function is not in the scope of this work.	Info
When details are being displayed, the respective button text must change from *Show Details* to *Hide Details*.	Must
When the *Hide Details* button is pressed, the count representation has to change back to the circular form (3) and the Button text has to change back to *Show Details*.	Must
All graphical components have to behave as designed in Figma with regard to styling,hovering, animations and so on.	Must
The length of the dropdown menu must be the minimum length that the json configuration allows as designed in Figma.	Must
When pressing the close button (6) the menu must close in an animated manner reverse to the way it opened.	Must
A click on a ship system must trigger a function *open_page* that gets the ship system name as argument.	Must

The function *open_page* is not defined further in this scope of work.	Info
Reusable components must be programmed for the system groups and the ship systems.	Must
Since the state information of the ship systems comes from the backend, the components must be able to handle globally managed state.	Must
The functionality of the graphic state representation must be done using global state as it could come from a centralized API.	Must