# DalleDress Specification

## Overall Structure

The application consists of the following main components and their relationships:

A screenshot of a computer

Description automatically generated

## Hierarchy of the Components

The hierarchy of the components is as follows:

Application

├─ Header

│  └─ AppStatus

│

├─ NavBar

│  └─ ServerStatus

│

└─ View

├─ CurrentView

│   ├─ ViewHeader

│   ├─ ViewMain

│   └─ ViewSidebar

└─ ViewStatus

This hierarchy outlines the nesting of components, showing their parent-child relationships. Each component is nested under its parent component according to the application's structure. Components are attached to a particular location on its parent and stay attached no matter the size or shape of the window. Components never overlap each other. Some components will never overlap (Header, ViewStatus, etc.) Some will (MainView, MainSidebar). If a component would overlap, instead it scrolls.

## Description of Each Component

* Application
  + **Purpose:** The root component that encapsulates the entire interface.
  + **Position:** Encompasses all other components and takes up the entire window. It never scrolls.
  + **Sizing:** 100% width and height of the window
  + **Attached:** All four corners of the window
* Header
  + **Purpose:** Provides an area to display the name of the application, the logo, and the AppStatus.
  + **Position:** Spans the full width at the top of the Application. It never scrolls and cannot be hidden.
  + **Sizing:** 100% width, 5% height of the Application
  + **Attached:** Top, Left, and Right of container
  + **AppStatus**
    - **Purpose:** Displays status information about the app such as currently opened file, time, date, etc.
    - **Position:** Located at the top-right corner of the Header.
    - **Sizing:** 10% width, 100% height of the Header
    - **Attached:** Top, bottom, and right of its container
* NavBar
  + **Purpose:** The navigation bar that allows users to navigate through different views of the application.
  + **Position:** Spans the full height on the left side of the Application exclusive of the Header. It never scrolls.
  + **Sizing:** 15% width and 95% height of the Application
  + **Attached:** Bottom and left sides of its container
  + **ServerStatus**
    - **Purpose:** Displays the status of the RPC (and perhaps File) server(s).
    - **Position:** Located at the bottom of the NavBar.
    - **Sizing:** 100% width and 5% height of the NavBar
    - **Attached:** Bottom, left, and right of the NavBar
* View
  + **Purpose:** A invisible placeholder to carry function for all views.
  + **Position:** All the area in the Application exclusive of the Header and the NavBar. The central area of the Application. Contains various views. The CurrentView covers its entire surface, making this component invisible to the user. This component never scrolls.
  + **Sizing:** 85% width and 95% height of its container.
  + **Attached:** Bottom and right sides of its container
  + **CurrentView**
    - **Purpose:** The primary container within View that holds the main content sections.
    - **Position:** Inside View, containing three sub-components: ViewHeader, ViewMain, and ViewSidebar.
    - **Sizing:** 100% width, 100% height of the View
    - **Attached:** Bottom right
    - **ViewHeader**
      * **Purpose:** The top section within CurrentView, which may be empty and therefore, while always present, may be invisible.
      * **Position:** Positioned at the top of CurrentView.
      * **Sizing:** When empty, 0% height and 100% width of CurrentView. When not empty, 5% height and 100% width of CurrentView
      * **Attached:** Top, left, and right of CurrentView
    - **ViewMain**
      * **Purpose:** The primary area to display information within CurrentView.
      * **Position:** Positioned in the center of CurrentView exclusive of the ViewHeader and ViewSidebar. Scrollable if its contents would otherwise overlay any other components inside the same container.
      * **Sizing:** 95% height of CurrentView when ViewHeader is empty, 90% height of CurrentView when ViewHeader is not empty. 90% width of CurrentView.
      * **Attached:** Bottom and left sides of its container
    - **ViewSidebar**
      * **Purpose:** May be empty. If not, it displays ancillary information for the CurrentView.
      * **Position:** Positioned on the right side within CurrentView if not empty, invisible otherwise. Scrollable like ViewMain.
      * **Sizing:** 95% height of CurrentView when ViewHeader is empty, 90% height of CurrentView when ViewHeader is not empty. 10% width of CurrentView when it itself is not empty. 0% if empty.
      * **Attached:** Bottom and right sides of the CurrentView.
  + **ViewStatus**
    - **Purpose:** Displays the status and progress related into for what’s happening on the CurrentView.
    - **Position:** Positioned at the bottom of the View component. Always visible.
    - **Sizing:** 5% height and 100% width of the View
    - **Attached:** Bottom, left, and right of View.

## UI Layout Details

This information is somewhat redundant. Use the above information as the primary source.

* **Application**
  + The Application component covers the entire window. It consists of the Header at the top, the NavBar on the left, and the View occupying the remaining space.
* **Header**
  + Positioned at the top of the Application, spans the full width (100%) and 5% of the height. The AppStatus is located at the top-right corner.
  + **AppStatus**
    - Positioned at the top-right corner of the Header, spans 10% of the width and 100% of the height of the Header.
* **NavBar**
  + Positioned on the left side of the Application, spans 15% of the width and 95% of the height. The ServerStatus is located at the bottom.
  + **ServerStatus**
    - Positioned at the bottom of the NavBar, spans 100% of the width and 5% of the height of the NavBar.
* **View**
  + Positioned in the remaining space of the Application, spans 85% of the width and 95% of the height. It contains the CurrentView which is positioned at the top and the ViewStatus which resides at the bottom.
  + **CurrentView**
    - Positioned inside the View, spans 100% of the width and 100% of the height of the View. It contains the ViewHeader, ViewMain, ViewSidebar.
    - **ViewHeader**
      * Positioned at the top of the CurrentView. When empty, spans 0% height and 100% width. When not empty, spans 5% height and 100% width of the CurrentView.
    - **ViewMain**
      * Positioned in the center of the CurrentView, spans 90% width. When ViewHeader is empty, spans 95% height. When ViewHeader is not empty, spans 90% height of the CurrentView.
    - **ViewSidebar**
      * Positioned on the right side within CurrentView, spans 10% width. When ViewHeader is empty, spans 95% height. When ViewHeader is not empty, spans 90% height of CurrentView.
  + **ViewStatus**
    - Positioned at the bottom of the View, spans 5% height and 100% width of the View.

## Functional Requirements

* **Application:** Acts as the root container for all other components. It ensures the layout and structure of the application, maintaining the 100% width and height of the window.
  + **Header:** Displays the name of the application and a logo. It remains fixed at the top of the Application.
    - **AppStatus:** Displays real-time status information such as time, date, and file status. It updates dynamically as needed.
  + **NavBar:** Provides navigation links for different sections of the application. It remains fixed on the left side of the Application. This component can be hidden or minimized and possibly have its size adjusted.
    - **ServerStatus:** Displays the status of any servers, such as the RPC connection, server health, or API servers. It updates dynamically based on server status changes.
  + **View:** Serves as the main content area of the application. It remains fixed in the center of the Application, containing the CurrentView which occludes it entirely.
    - **CurrentView:** Acts as the primary container within the View, holding the main content sections. It dynamically adjusts the visibility and size of its sub-components based on their content.
      * **ViewHeader:** Displays a header within CurrentView. It can be empty, and its visibility and size adjust accordingly.
      * **ViewMain:** Displays the central content within CurrentView. It is scrollable if its contents exceed the available space, ensuring that all content is accessible.
      * **ViewSidebar:** Provides additional content or navigation options on the right side of CurrentView. It is scrollable like ViewMain and adjusts its size based on ViewHeader visibility. It can be empty, and its visibility and size adjust accordingly.
    - **ViewStatus:** Displays status information related to the main view content, such as progress or state indicators. It remains fixed at the bottom of the View and updates dynamically.