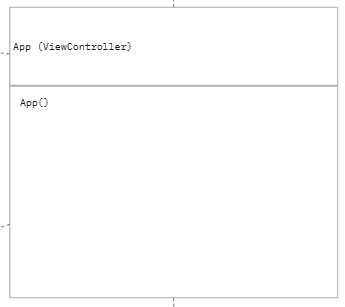
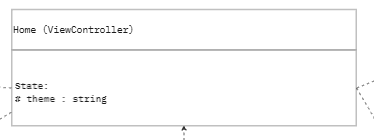
**TeamC Design “Class” Diagram**

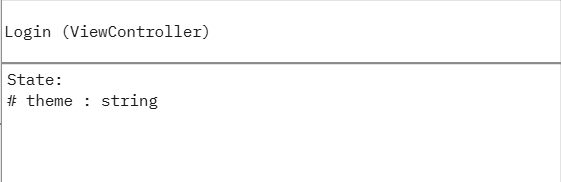
<https://miro.com/app/board/uXjVLCyKBRo=/>



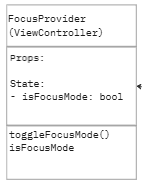
The diagram shows the base of our program (app) which everything on the website is displayed from.



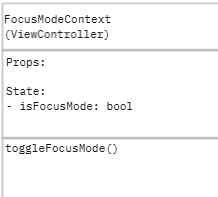
The diagram Home is pulling from App, giving access to both Focus and DarkMode Providers. Home uses FocusModeContext and DarkModeContext to change the “theme” or “isFocusMode” values, changing the way it’s displayed.

**

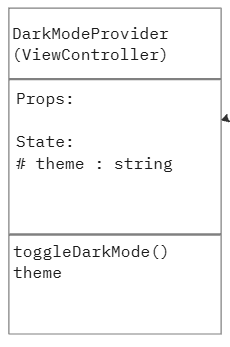
This diagram is of the function Login where the log in page information is stored. The display changes state depending on the theme state derived from DarkModeContext.



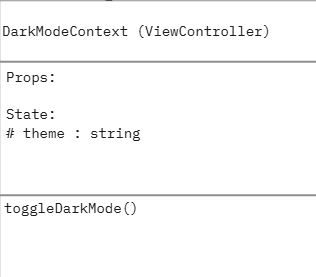
The diagram shows a state FocusProvider which is being used by App to access it’s information. FocusProvider is a gateway that lets other files access its isFocusMode state as well as the ability to toggle it with toggleFocusMode().



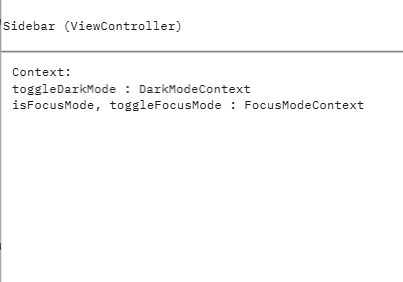
The diagram shows the FocusModeContext which manages and shares the isFocusMode state across components. It also includes a toggle function to enable and disable focus mode.



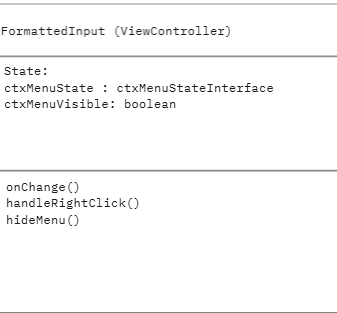
The diagram shows a state DarkModeProvider which is being used by App to access it’s information. DarkModeProvider is a gateway that lets other files access its theme state as well as the ability to change it with toggleDarkMode().

**

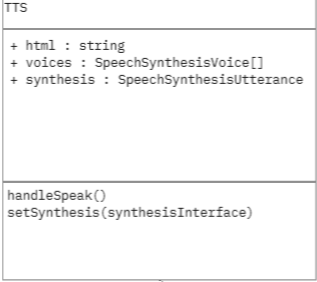
The diagram shows the DarkModeContext which manages and shares the “light/dark” across components. It provides a theme state and a toggle function to switch between the two. It uses the DarkContextProvider to make the theme accessible globally.

**

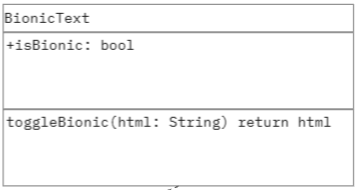
This diagram shows the Sidebar function that displays the sidebar and its many elements. Sidebar is derived from Home which carries information from DarkModeContext and FocusModeContext. The display of the sidebar changes depending on the current state of either “theme” from DarkModeContext, or “isFocusMode” from FocusModeContext.

**

This diagram is for the text box on the homepage that is used for text reformatting. It opens a context menu when it is right clicked, and closes the context menu when it is left clicked.



This class diagram is for the TTS feature which has controls for volume and speed, which are set by setSynthesis when the value of a slider element is changed.



This diagram is for the Bionic Text feature which is toggled on and off with a button to reformat text in the Editor.

**Design Patterns**

**State design pattern**

We used the State design pattern throughout the architecture of our website for multiple different elements. We used the state setter for toggling on and off Dark Mode by pressing the Toggle Dark Mode button on the sidebar. Once clicked, the theme (state) of the website would change to a black or white background and persist throughout any page. We later added a Focus Mode where pressing the Hide/Show All button would toggle visibility of the entire sidebar, again persisting throughout all pages. The state and state setter is used in multiple parts of the application to change the appearance of UI elements. This is important for our design because user action changes the state of an object.

