Boeing Roadmap Project Design

Full Design (Desktop):

Components

LabelList

This component is a instances of “AppLabel” components. It essentially serves simply as a container for the AppLabels and has the following functionality:
1. Must accept an array of objects with the following syntax:

```javascript
{
    "name": String <someName>,
    "Color": String <someColor>,
    "startDate": Date <someDate>,
    "endDate": Date <someDate>
}
```

2. Must generate number of AppLabel objects and display on Home template
3. Must have flexbox css property to allow it to be mobile responsive
4. Must communicate with bar-container component, which shares the following properties:
   a. ApplicationList bar height is equal to bar-container bar height
   b. ApplicationList bar colors should be equal to BarView bar-colors (object?)
5. This element is not scrollable
6. Mouse wheel down should pan the list down if possible. If not, give user visual cue that the list cannot be scrolled down further
7. Mouse wheel up should pan the list up if possible. If not, give user visual cue that the list cannot be scrolled down further.
8. Should have order function, which sorts the AppLabel objects in one of the following orders:
   a. Alphabetical [A-Z]
   b. Alphabetical [Z-A]
   c. Most Recently Introduced
   d. Most Recently Retired
9. Should run order(0) on initialization

```
OrderDropDown
```

The order dropdown allows the user to select an order by which the LabelList component arranges the AppLabel elements. By default, this has the placeholder text "Alphabetical [A-Z]" and lists the AppLabel components as specified. More descriptively, this component should:

1. Call LabelList’s order function with a parameter that reflects all possible selections.
   a. order(0) should take the objects described in LabelList and apply some alphabetical sorting algorithm using the object’s name property
   b. order(1) should take the objects described in LabelList and apply some reverse alphabetical sorting algorithm using the object’s name property
   c. order(2) should take the objects described in LabelList and apply some date sorting algorithm using the object’s startDate property and today’s date
   d. order(3) should take the objects described in LabelList and apply some date sorting algorithm using the object’s endDate property and today’s date
2. Lifecycle hook will be “onChange”
This component is a timeline of the roadmap application. It displays a month to month view so that users can see when applications begin and end. It interacts with our DatePicker to determine the time frame to display. It’s detailed functions are:

1. Should only display the dates in the inclusive range of the earliest introduced application date and the latest application expiration date. This information will be served via the `getData` directive mentioned later in this document.
2. Contrary to the design layout, the timeline should not have arrows, as panning by middle-mouse clicking and scrolling will suffice (NOTE: Must test on Macs, what is the 

   *Apple equivalent to middle mouse click/scroll?*).
3. As internal zoom state is increased, there should be more space allocated between the timeline points of reference.
4. The lines for the BarView must be placed behind the actual BarView so that it is legible, but not intrusive. This can be achieved with a Z-Index CSS Property

This component will display the instances from our AppLabel component. It will be scrollable vertically, horizontally and have zooming capabilities (see Timeline component’s 2 specification).

1. Should have Z-Index higher than Timeline to allow lines to fall behind it
2. (Proposed Design) container areas in a “hamburger” to ensure no overlaps in bars
3. Should get the following information from the LabelList Component

   a. Color, to match the set AppLabel component color
   b. StartDate, to set starting point of BarElement
   c. EndDate, to set end point of BarElement
**DatePicker**

This element sets the endpoints of the Timeline and BarView components. Suggested specifications:
1. Should only set the endpoints when both “From” and “To” inputs have valid data.
2. Should express invalid input with reason printed under the component in red text

**Checkboxes**

Removes or adds AppLabel Objects to the LabelList component when checked or unchecked:
1. Show **Deprecated** Apps should add/remove all LabelList objects that have endDates earlier than today’s date
2. Show Living Apps should add/remove all LabelList objects that have StartDates earlier than today’s date && endDates later than today’s date
3. Show Upcoming apps should add/remove all LabelList objects that have startDates later than today’s date

**Search**

Search based off Name:
1. The String from user input is translated to lowercase via pipe
2. System uses lowercase String to search LabelList component
3. Exports results based on String to be used in BarView to display Application

Search based off Keyword: (optional)
1. The String from user input is translated to lowercase via pipe
2. System uses lowercase String to search LabelList component for specified “keyword”
3. Exports results based on keyword to be used in BarView to display Applications
Export:

1. Populate **AppList: startDate-endDate**
2. Generate .pdf using populated data

**HeaderContainer**
Contains the Timeline and OrderDropDown components. This element should have a set minimum height so that is always readable, even if the height of the browser is relatively small.

**MainContainer**
Contains the LabelList and BarView components. This element should shrink and grow to width changes, but not to height changes. This is because these elements should be scrollable.

**FooterContainer**
Contains the DatePicker, CheckBoxes, Search, and Export components. This component should have a width media query to make it so that when the browser is so small that the MainContainer isn’t big enough, the FooterContainer becomes accessible through a button (or possibly a pull-up menu).

**Directives**

**DataWorker**
This directive pulls in data from the data source and exports an array of objects with the following syntax:

```json
{
    "name":String <someName>,
    "Color":String <someColor>,
    "startDate": Date <someDate>,
    "endDate": Date <someDate>
}
```

**CSVConverter**
Reads from the Boeing Application .csv file and converts each column into a readable format.
Classes

AppLabel

PROPERTIES:
- String name
- String color
- Date startDate
- Date endDate

Constructor ( name:string, color:string, startDate:Date, endDate:Date )

FUNCTIONS:
- Property Accessors for name, color, startDate, endDate
- Property Mutator for color