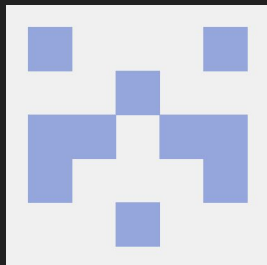


CS4530 Final Project

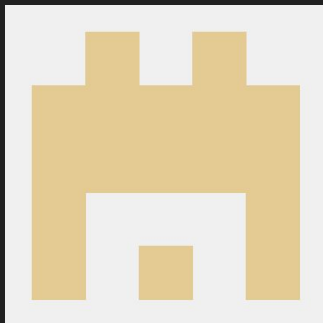
By: Hector Padilla, Ousman Jobe, Madison Lebiedzinski, Dylan Mccann

Introduction - Team 109!

Hector Padilla



Ousman Jobe



Maddie Lebiedzinski



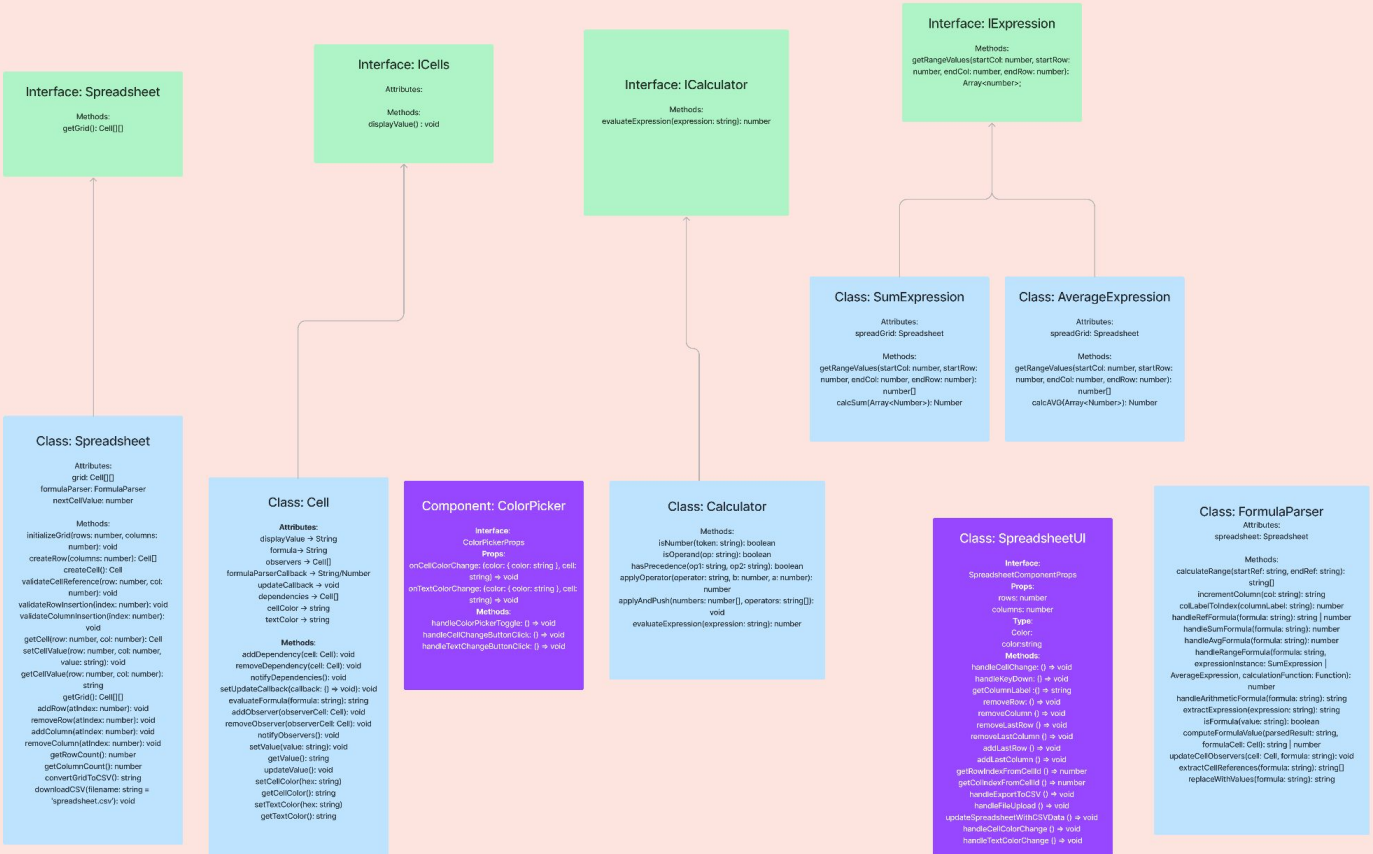
Dylan McCann



Demo

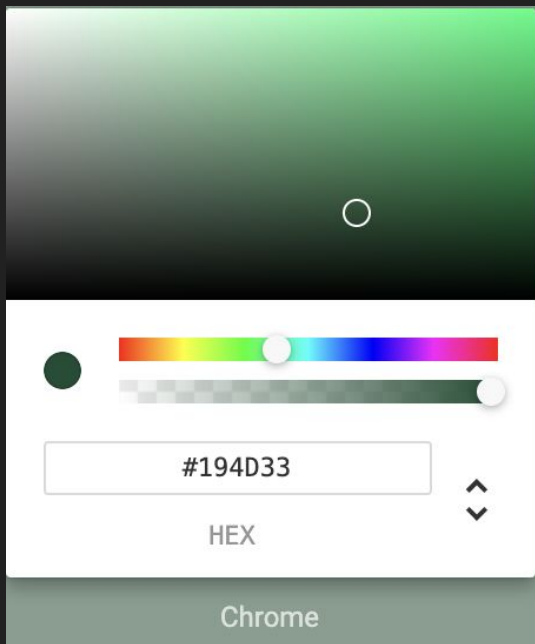


System Architecture



Code Snippet

```
import { ChromePicker, ColorResult } from 'react-color';
```



```
const ColorPicker: FC<ColorPickerProps> = ({ onCellColorChange, onTextColorChange }) => {  
  const [color, setColor] = useState<string>("#fff");  
  const [showColorPicker, setShowColorPicker] = useState<boolean>(false);  
  const [showAdditionalButton, setShowAdditionalButton] = useState<boolean>(false);  
  
  const handleColorPickerToggle = () => {  
    setShowColorPicker(prevState => !prevState);  
    setShowAdditionalButton(prevState => !prevState);  
  };  
  
  const valueRef = useRef<HTMLInputElement>(null);  
  
  // Sends the color and specific cell from the color picker to update in spreadsheet component  
  const handleCellChangeButtonClick = () => {  
    if (valueRef.current) {  
      onCellColorChange({ color }, valueRef.current.value);  
    }  
  };  
  
  const handleTextChangeButtonClick = () => {  
    if (valueRef.current) {  
      onTextColorChange({ color }, valueRef.current.value);  
    }  
  };  
  
  return (  
    <div>  
      <button onClick={handleColorPickerToggle}>  
        {showColorPicker ? 'Close color picker' : 'Pick a color'}  
      </button>  
  
      {showColorPicker && (  
        <ChromePicker  
          color={color}  
          onChange={({ updatedColor: ColorResult } => setColor(updatedColor.hex))  
        />  
      )}  
    </div>  
  );  
};
```

Lessons Learned

- What went well
 - Work Scheduling
 - Our sprint was at a nice pace where we were able to consistently get user stories implemented and finish with time before the deadline
 - Teamwork and Communication
 - Everyone worked well together, consistently communicating and contributing
 - UML, Cell References, Calculator
- What didn't go well
 - .js to .ts
 - We accidentally set up our project for js instead of ts and a couple times accidentally wrote js code and had to convert it to ts which caused some unnecessary work and hassle.
 - Cell Dependencies:
 - Our initial approach to implementing the cell dependencies was flawed in that it would not correctly update dependent cell values, which led to refactoring and was overall challenging.
- What we would do differently
 - Parser
 - We have implemented our own parser that works well, however there are third party parsers that could be implemented and definitely improve on the scope and performance of our parsing
 - Selecting Cells
 - We ended up altering our initial plan for selecting cells such that it works but is different than the traditional way that mainstream spreadsheets use which is probably the best implementation of it.