
Spreadsheet 509

— Nhan Son, Arihant Pant, Minh Le, —
Nikolas Brisbois

Scope of the Presentation

- Demonstration of all functionalities
- Insight into the system's architecture and the technical considerations behind its development
- Highlighting an interesting code snippet
- Reflection on the lessons learned throughout the development process, including successes, challenges, and areas for future improvement.

Must-Have Features

The spreadsheet supports:

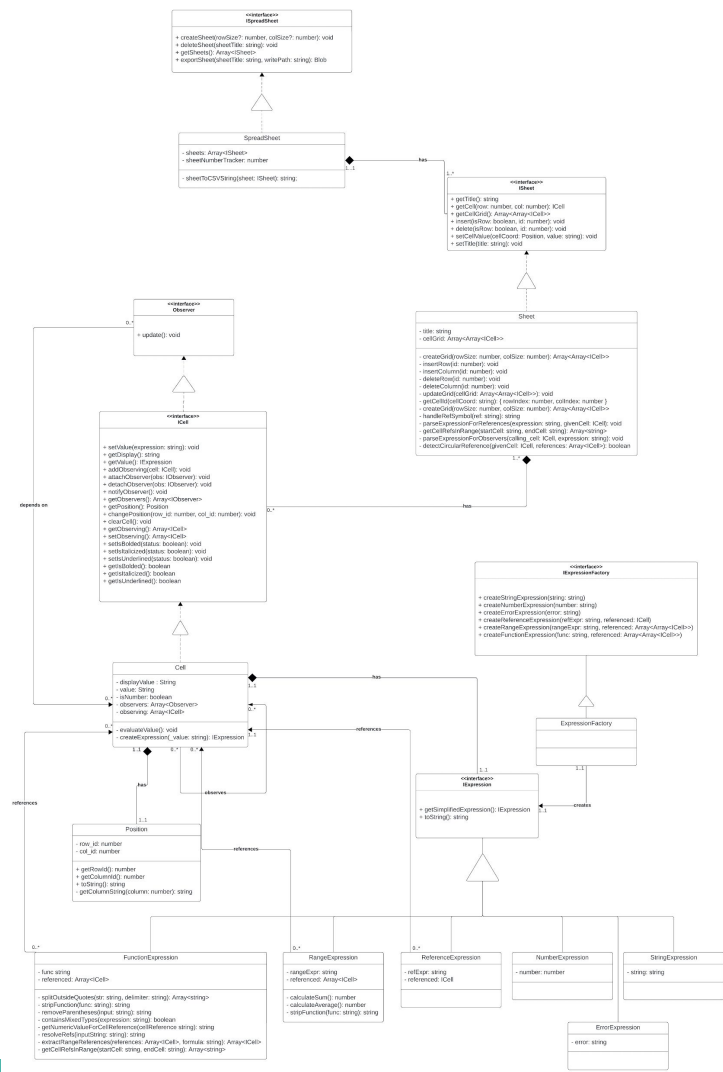
- Versatile Cell Content
- Formulas
- String Concatenation
- Cell Reference and Ranges
- Error Detection
- Sheet Management: Insert and delete rows and columns, clear cells
- Interactive Web Interface: Developed using React, providing real-time data manipulation and display.

Additional Features

- Font Formatting: Format the cell to be bold, italic, and/or underlined
- Multi-Sheet Management: Handle multiple sheets within a single spreadsheet.
- CSV Export: Export sheets to CSV format – compatible with other spreadsheet applications

System Architecture

- Spreadsheets are a container for Sheets
- Sheets are a container for Cells
 - Assigns observers and dependencies to cells
- Cells represent their values as an expression
 - Can be an observer and/or a dependent
 - Expressions may be simplified



Interesting Code Snippet

String Concatenation

Current Sheet: 0

0

⋮

A2

=(REF(A1)+REF(B1))

SAVE

CLEAR

B

I

U

	A	B	C	D	E	F	G	H
1	"(1+2)"	"=3"						
2	(1+2)=3							
3								
4								
5								
6								
7								
8								
9								

ADD SHEET

Current Sheet: 0

0

A2

=(REF(A1)+REF(B1))

SAVE

CLEAR

B

I

U

	A	B	C	D	E	F	G	H
1	cat	"dog44"						
2	catdog44							
3								
4								
5								
6								
7								
8								
9								

ADD SHEET

```

// 2 days ago - Find ways to include special character
// Check if the expression is a concatenation of pure strings or strings in quotes, separated only by +
if (/^\(((\s*"([^"]|""\s*)"|\s*[a-zA-Z]+\s*)+(\s*(\s*"([^"]|""\s*)"|\s*[a-zA-Z]+\s*)+)*\)$/.test(modifiedExpression)) {
    modifiedExpression = this.removeParentheses(modifiedExpression);

    // Use the custom split function
    modifiedExpression = this.splitOutsideQuotes(modifiedExpression, '+').map(s => {
        s = s.trim();
        if (/^".*"$/i.test(s)) {
            return s; // Return as is if already in quotes
        }
        return `"${s}"`; // Wrap in quotes otherwise
    }).join(',');
    modifiedExpression = `CONCAT(${modifiedExpression})`;
}

```

```

// Define the CONCAT function
parser.setFunction('CONCAT', (params: unknown) => {
    if (!Array.isArray(params)) {
        throw new Error('Expected an array');
    }
    if (!params.every(param => typeof param === 'string')) {
        throw new Error('Expected all parameters to be strings');
    }
    return params.join('');
});

```

Lesson Learned

Successes:

- Base features and additional features are all implemented and are functioning
- Code is documented and tested
- Spreadsheet is responsive

Challenges:

- Had to scrap some of our additional features in favor of others
 - Imports
 - Graphs
- Did not have a consistent work schedule, most of the work was towards the last few weeks

What we would do differently:

- Figure out story points properly so we can have a better gauge of what's easy and hard to implement
- Do more code reviews

Thank You

Questions?