

Team 108 Project Phase B Deliverables:

User Stories:

TITLE: Create a spreadsheet (US-01)

PRIORITY: high

ESTIMATE: 2

AS A user

I WANT TO use the application to create a new, empty spreadsheet

SO THAT I can start adding data and utilizing the features of the program.

ACCEPTANCE CRITERIA:

- The user can perform an action to create a new spreadsheet
 - An empty spreadsheet is created
 - Multiple spreadsheets can be created and saved at the same time
 - The user should be able to name spreadsheets as they are being created
 - The user should be able to specify the size or type of spreadsheet during its creation
-

TITLE: Click on a cell (US-02)

PRIORITY: high

ESTIMATE: 1

AS A web user

I WANT TO be able to use my mouse to click on a cell in a spreadsheet

SO THAT I can select, edit, view, or delete the data in the cell that I click on

ACCEPTANCE CRITERIA:

- User can use their mouse pointer to click on a cell
 - Once clicked, the cell will be selected and the formula or data displayed
 - Once clicked, the cell can be edited
 - Once clicked, the cell is highlighted to represent that it is selected
-

TITLE: Enter characters (US-03)

PRIORITY: high

ESTIMATE: 1

AS A user

I WANT TO use my keyboard to enter a character into the spreadsheet

SO THAT I can enter and edit data in the spreadsheet easily

ACCEPTANCE CRITERIA:

- User can use their keyboard to input characters into cells in the spreadsheet
 - Characters appear on the screen according to the UI design
 - Any supported character can appear as expected
 - Any unsupported characters will not appear on the spreadsheet and will not produce a major error that disrupts functionality
-

TITLE: Cell clearing (US-04)

PRIORITY: high

ESTIMATE: 0.5

AS A spreadsheet user I WANT TO clear the contents of a cell so that it is empty again SO THAT if I need to make room for new calculations or fix mistakes, I can simply remove whatever is currently in the cells.

ACCEPTANCE CRITERIA:

- Be able to click on a cell and press some button to remove the equation from the cell and reset it to its default state
 - Once a cell has been cleared, it has identical functionality to a brand new cell
 - Cells that reference the cleared cells are handled properly to reflect the deletion of data
-

TITLE: Delete row/column (US-05)

PRIORITY: high

ESTIMATE: 2

AS A user

I WANT TO delete a row or column from an existing spreadsheet

SO THAT if a column is unnecessary or a spacer column it can be easily discarded without having to manually move other rows. I want to reduce the data per row, I want to be able to delete unwanted columns.

ACCEPTANCE CRITERIA:

- User can delete specific rows and columns from the spreadsheet
 - Remaining rows and columns refactor themselves as necessary
 - Once a row is deleted, the remaining row/cols have no bugs or visual glitches
 - A user should be asked if they are sure they want to delete a row/col if it has data in it
-

TITLE: Insert row/column (US-06)

PRIORITY: high

ESTIMATE: 2

AS A user

I WANT TO add an empty row or column to an existing spreadsheet

SO THAT if I need additional space within part of the spreadsheet, I can add additional rows or columns of blank space. This will allow me to insert new fields into the dataset, which will allow for analysis of more parameters.

ACCEPTANCE CRITERIA:

- User can add empty rows and columns from the spreadsheet
 - Other rows and columns refactor themselves as necessary
 - An empty row/col contains the correct number of corresponding empty columns/rows
 - Empty row/col functionality is fully featured and has no limitations or bugs
-

TITLE: Formulas (US-07)

PRIORITY: medium

ESTIMATE: 2

AS someone interested in applying mathematical operations in a spreadsheet

I WANT TO write formulas in the cells of a spreadsheet of data in a readable and usable way and to have the tools at my disposal to view and update formulas quickly.

SO THAT I can perform complex calculations without using a calculator. This will allow me to work quickly when I am applying complex column transformations.

ACCEPTANCE CRITERIA:

- The user can enter formulas into a spreadsheet
 - At a minimum should support (+, -, *, /, ^)
 - Formulas are evaluated correctly and the results are given to the user
 - When mathematical formulas are able to be entered, correctly parsed, and applied to the cell they are in.
 - When a cell displays its formula when clicked on.
 - When a cell's formula can be updated or changed simply by typing when the cell is selected.
 - When formula syntax is easily readable, and documentation exists if necessary.
-

TITLE: Parentheses (US-08)

PRIORITY: medium

ESTIMATE: .5

AS A mathematician

I WANT TO be able to use parentheses in my calculations in a spreadsheet

SO THAT I can input complex expressions using proper order of operations

ACCEPTANCE CRITERIA:

- A user can enter parentheses into a cell surrounding mathematical expressions
 - Expressions are now evaluated preserving order of operations such that terms inside parens are evaluated first
-

TITLE: Reference cells (US-09)

PRIORITY: high

ESTIMATE: 1

AS A data analyst

I WANT TO be able to reference the data from other cells from within a cell in the spreadsheet
SO THAT as values within the spreadsheet change, calculated values can automatically update
to remain accurate.

ACCEPTANCE CRITERIA:

- A user can use a reference of any cell in the spreadsheet within a cell
 - The data referenced is properly translated and updated as needed
 - User can enter a reference to the value within another cell and have it populate in the equation
 - When the referenced cell's value is changed, this cell automatically updates it's value as well
-

TITLE: Range expressions (US-10)

PRIORITY: low

ESTIMATE: 2

AS A spreadsheet user I WANT TO be able to take the sum or average over a range of cells
within the equation for a cell SO THAT taking the average or sum of a range of cells can be
simplified and each cell does not have to be referenced individually.

ACCEPTANCE CRITERIA:

- A cell can contain a single call to take the sum or average of a range of cells and will automatically update if any of those cells are updated
 - When the user is able to use both aggregation methods (SUM, AVG).
 - When the user is able to apply these aggregation methods to a range of values in one or more columns.
-

TITLE: Web Application (US-11)

PRIORITY: high

ESTIMATE: 5

AS A typical user

INSTEAD OF downloading a desktop application

I WANT TO open the spreadsheet application via a web browser

SO THAT I can use the application's features without downloading an .exe

ACCEPTANCE CRITERIA:

- User can run the application on any supported web browser
 - All features of the application work as expected on the browser
 - The application is responsive and interactive as expected
 - The application has been tested for errors and bugs
-

ADDITIONAL FEATURES BELOW:

TITLE: *Import dataset from file (US-12)*

PRIORITY: high

ESTIMATE: 2

AS A data analyst

INSTEAD OF manually entering data to the spreadsheet

I WANT TO import data from a CSV(or other type) file

SO THAT I can easily work with existing datasets

ACCEPTANCE CRITERIA:

- The user can select and import data from existing CSV files
 - Other file types may be included but are not necessary for acceptance
 - The data is imported into a new spreadsheet, or the user is given the option to enter the data into the existing spreadsheet
-

TITLE: *Export spreadsheet to file (US-13)*

PRIORITY: medium

ESTIMATE: 2

AS A desktop user

I WANT TO export the spreadsheet to a file on my computer (CSV or otherwise)

SO THAT I can save and share my dataset in other applications

ACCEPTANCE CRITERIA:

- The user can export the current spreadsheet to their computer as a CSV
 - Other file formats may be selectable but are not necessary for acceptance
 - The exported file is saved with the correct data onto the user's system
-

TITLE: *Undo/Redo Changes* (US-14)

PRIORITY: medium

ESTIMATE: 2

AS An everyday user

I WANT TO have the ability to undo/redo changes easily in a spreadsheet and restore the spreadsheet to its state before the action was taken

SO THAT I can fix my mistakes or it can be easily undone without having to manually restore the state before the action was taken to go back to previous versions of data.

ACCEPTANCE CRITERIA:

- User can perform an action to undo the last change made to the spreadsheet
 - User can perform a different action to redo the last undone change
 - Must be able to easily undo at least one user action, but hopefully up to 3 actions
 - Must be able to easily redo at least one undone user action, but hopefully up to 3 actions
-

TITLE: *Create charts* (US-15)

PRIORITY: medium

ESTIMATE: 5

AS An analyst

I WANT TO create visual charts or graphs from the data in a spreadsheet

SO THAT I can visualize data trends and patterns from the set.

ACCEPTANCE CRITERIA:

- The user can create various types of charts from a selection of data
 - Charts are communicated to the user in a useful way
 - At a minimum, there can be bar charts, line charts, and pie charts. With additional chart types being stretch goals.
 - Charts are customizable by color, and labels are editable
 - Charts can be saved alongside the current spreadsheet
-

TITLE: *Statistical Analysis* (US-16)

PRIORITY: medium

ESTIMATE: 3

As a data analyst, I want to gain a deeper understanding of the significant trends and relationships within the spreadsheet, instead of relying on more surface-level summaries such as the sum and average of a range of values. Specifically, I want to be able to conduct t-tests, correlations, and linear regression. Applying statistical methods will allow me to hone in on the

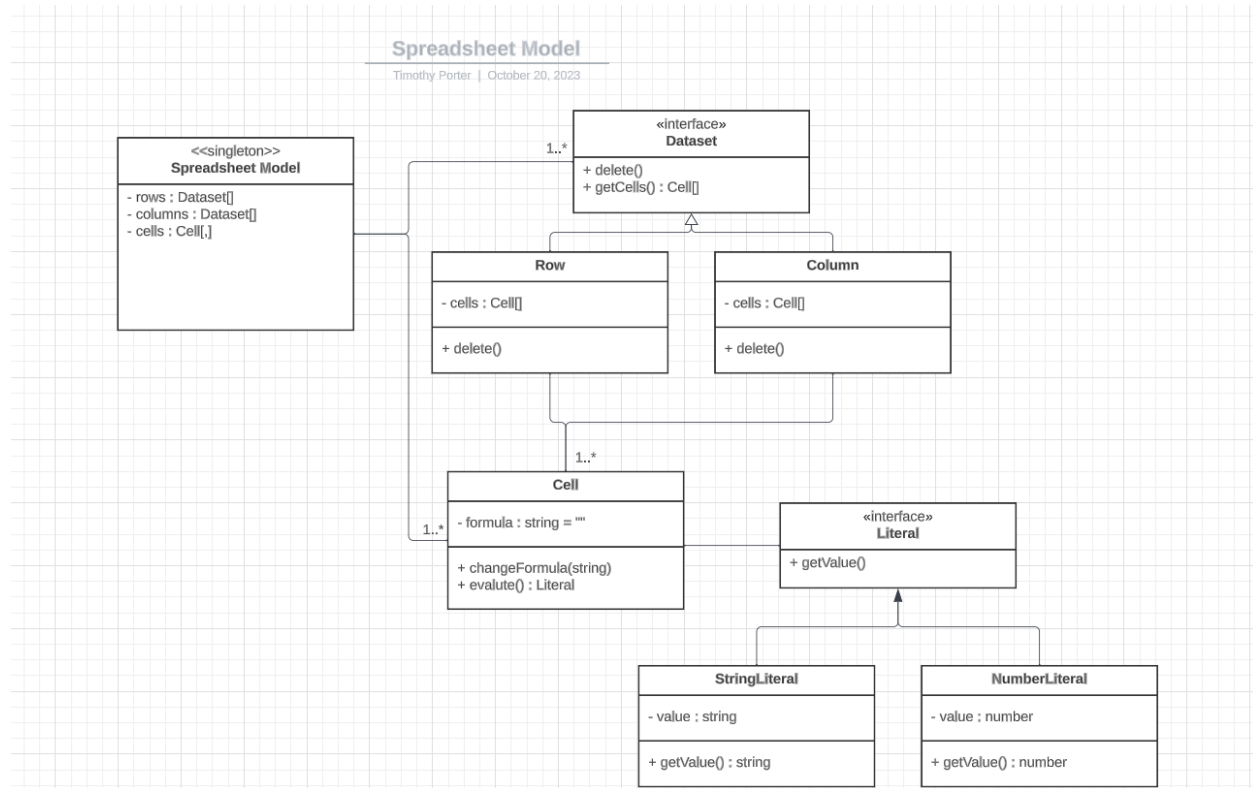
meaningful aspects of the data, which can be used to prepare visualizations for an external audience, as discussed in *Publishing Preparation*.

ACCEPTANCE CRITERIA:

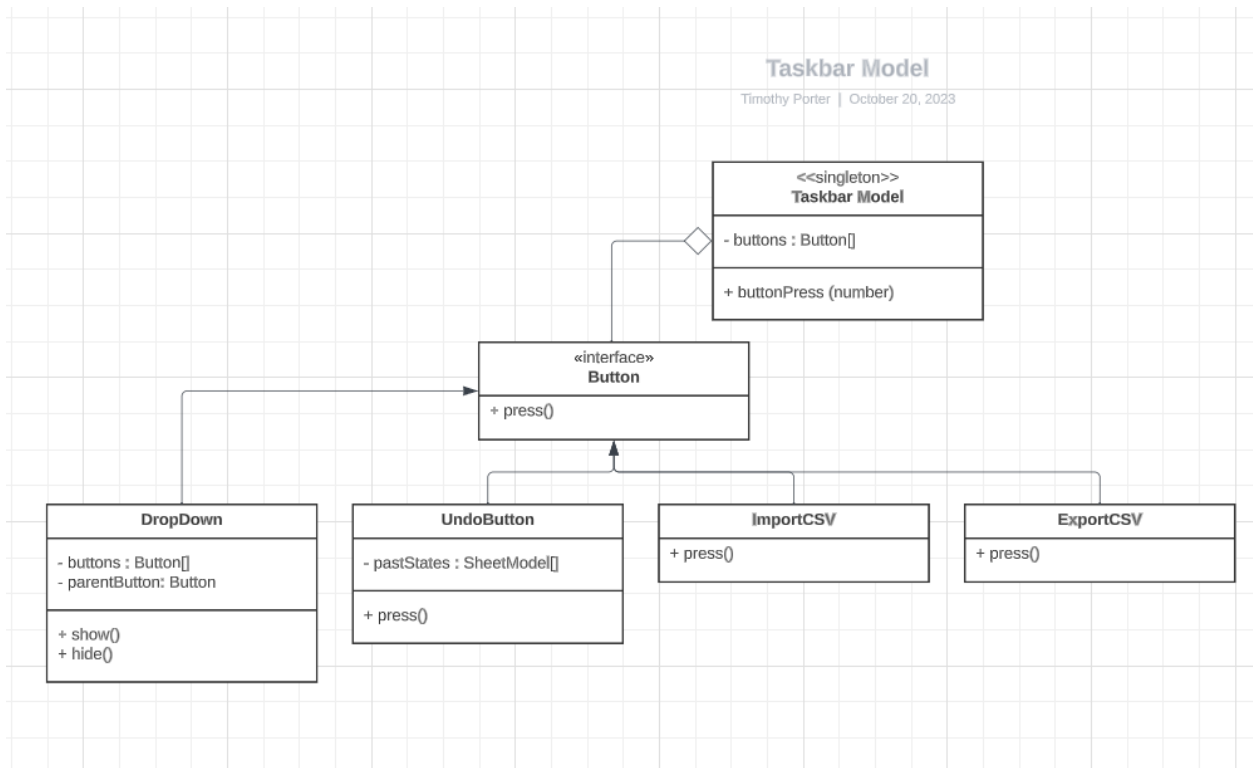
- When it is possible to conduct a t-test between two columns.
- When it is possible to conduct a t-test to find the difference in one column between two groups (see *Grouping Operations*).
- When it is possible to find the correlation (Pearson's R) between two columns.
- When it is possible to construct a simple linear regression of one variable predicting another variable.
- When it is possible to construct a multiple linear regression of multiple variables predicting one variable.
- When these linear regressions output a table showing statistics including coefficient, degrees of freedom, t-value, and p-value for each model parameter.
- When there are restrictions and error checks on statistical models. For example, a model cannot be applied to a range of empty cells. Furthermore, these models cannot be applied to a column containing strings.

UML Class Diagrams:

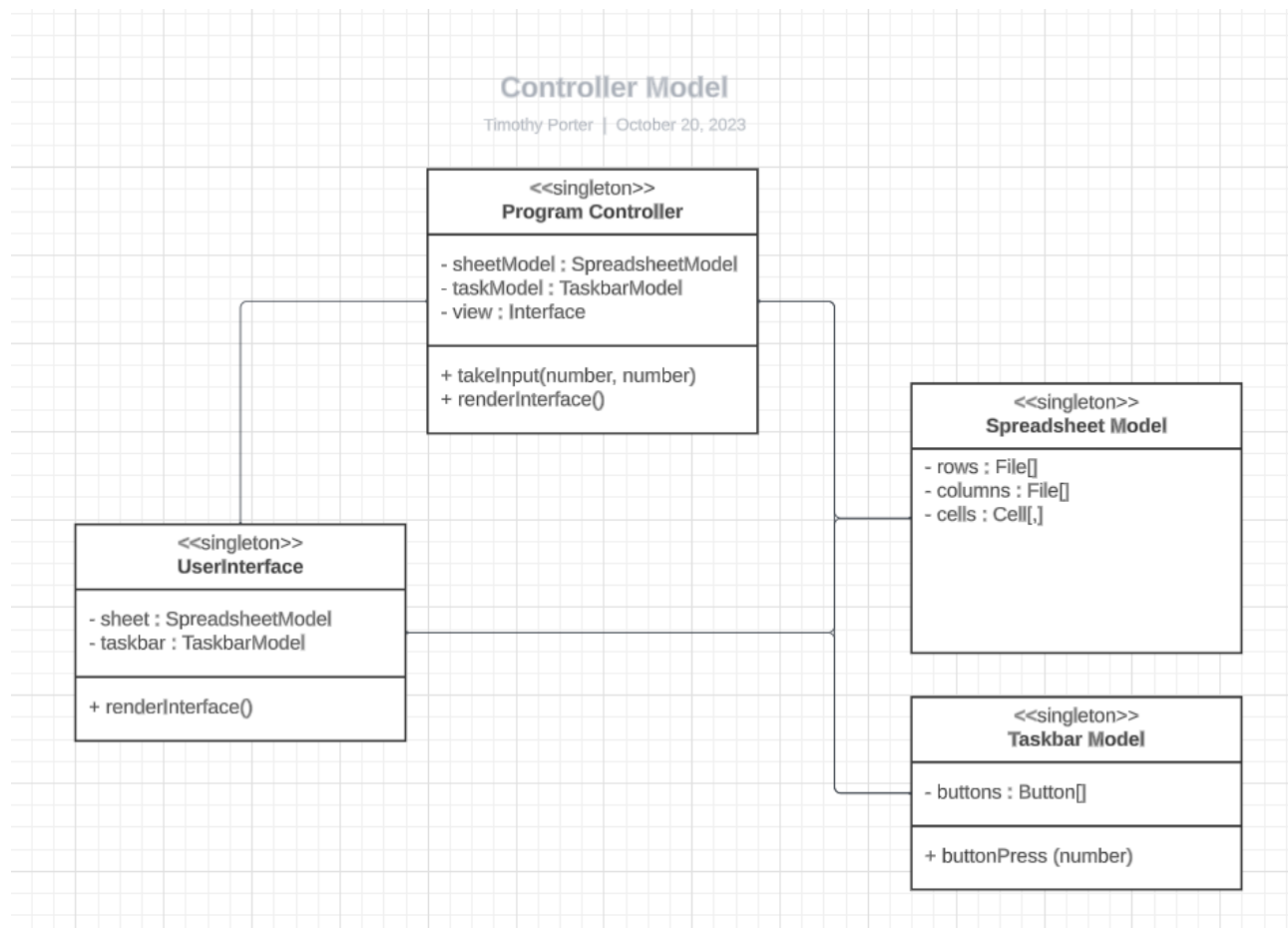
Spreadsheet Component Model



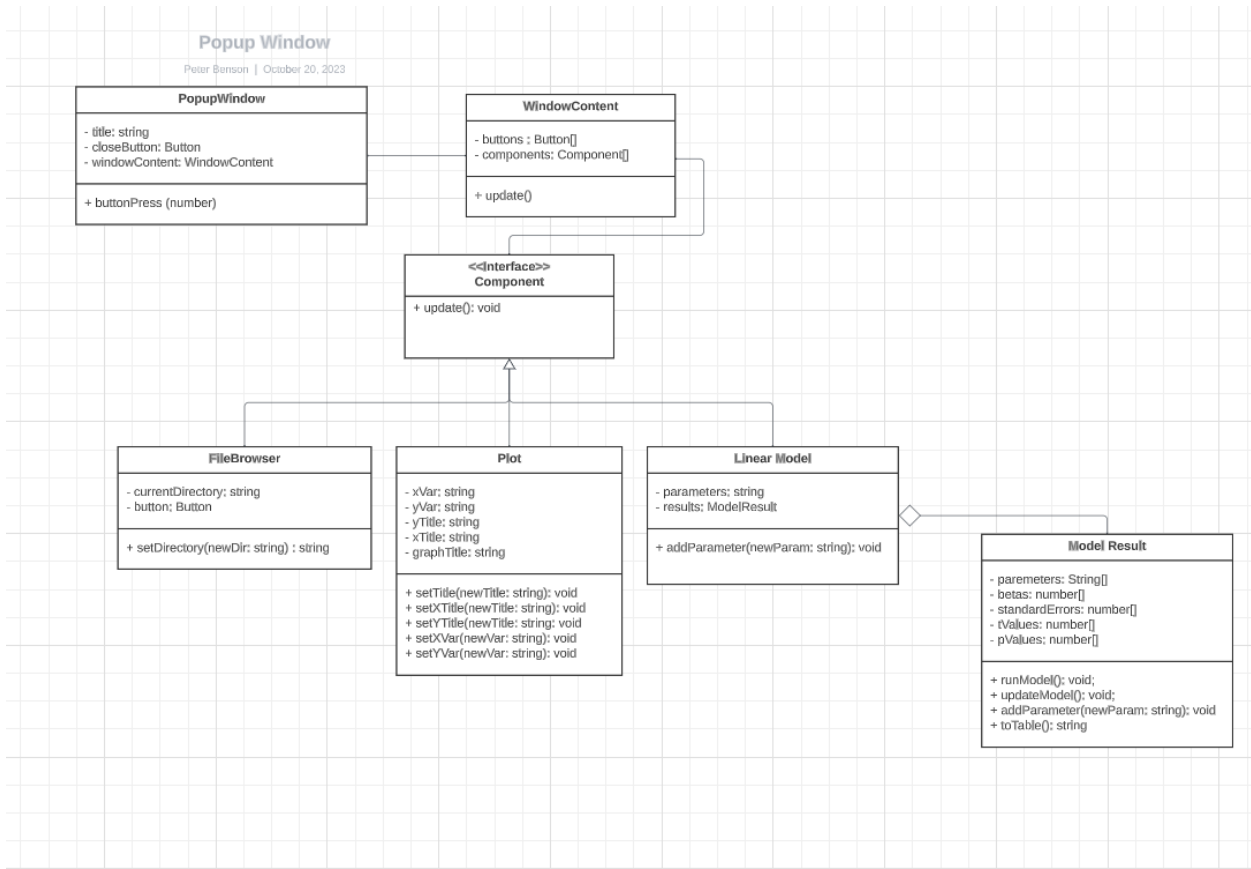
Taskbar Component Model



Unified controller model



PopupWindow classes



CODE:

```
export interface Button {  
    press(): void;  
}
```

```
export interface Component {  
    update(): void;  
}
```

```
export interface Dataset {  
    delete(): void;  
    getCells(): Array<Cell>;  
}
```

```
export interface Literal {  
  
}
```

```
export class Cell {  
    private formula: string;  
  
    public constructor() {  
        this.formula = '';  
    }  
  
    public setformula(string): void {  
        throw Error('Not implemented');  
    }  
  
    public getformula(string): string {  
        return this.formula;  
    }  
  
    public evaluate(): Literal {  
        throw Error('Not implemented');  
    }  
}
```

```
export class Column implements Dataset {  
    private cells: Array<Cell>;  
  
    public delete(): void {  
        throw new Error("Method not  
implemented.");  
    }  
    public getCells(): Cell[] {  
        return this.cells;  
    }  
}
```

```
export class DropDown implements Button {  
    private buttons: Array<Button>;  
    private parentButton: Button;  
  
    press(): void {  
        throw Error('not implemented.');    }  
  
    public show(): void {  
        throw Error('not implemented');  
    }  
  
    public hide(): void {  
        throw Error('not implemented');  
    }  
}
```

```
export class ExportCSV implements Button {  
  
    press(): void {  
        throw Error('not implemented.');    }  
}
```

```
export class FileBrowser implements Component {  
    private currentDirectory: string;  
    private button: Button;  
  
    public setDirectory(newDir: string): void {  
        this.currentDirectory = newDir;  
    }  
}
```

```
export class ImportCSV implements Button {  
  
    press(): void {  
        throw Error('not implemented.');    }  
}
```

```

    }

    public getDirectory(): string {
        return this.currentDirectory;
    }

    update(): void {
        throw new Error("Method not
implemented.");
    }
}

```

```

export class NumberLiteral implements Literal {
    private value: number;

    constructor(value: number) {
        this.value = value;
    }

    public getValue(): number {
        return this.value;
    }
}

```

```

export class ModelResult {
    private parameters: Array<string>;
    private betas: Array<number>;
    private standardErrors: Array<number>;
    private rValues: Array<number>;
    private pValues: Array<number>;

    public runModel(): void {
        throw Error('not implemented');
    }

    public updateModel(): void {
        throw Error('not implemented');
    }

    public addParameter(newParam: string): void {
        throw Error('not implemented');
    }

    public toTable(): string {
        throw Error('not implemented');
    }
}

```

```

export class LinearModel implements Component {
    private parameters: string;
    private results: ModelResult;

    public addParameter(newParam: string): void {
        throw Error('not implemented');
    }

    update(): void {
        throw new Error("Method not
implemented.");
    }
}

```

```

export class Plots implements Component {
    private xVar: string;
    private yVar: string;
    private xTitle: string;
    private yTitle: string;
    private graphtitle: string;

    public update(): void {
        throw new Error("Method not
implemented.");
    }

    public setXVar(newXVar: string): void {
        this.xVar = newXVar;
    }

    public setYVar(newYVar: string): void {
        this.yVar = newYVar;
    }

    public setXTitle(newXTitle: string): void {
        this.xTitle = newXTitle;
    }

    public setYTitle(newYTitle: string): void {
        this.yTitle = newYTitle;
    }

    public setGraphTitle(newGraphTitle: string):
void {
        this.graphtitle = newGraphTitle;
    }
}

```

```
export abstract class PopupWindow {
    title: string;
    closeButton: Button;
    windowContent: WindowContent;

    public buttonPress(): number {
        throw Error('not implemented');
    }
}
```

```
export class ProgramController {
    private sheetModel: Spreadsheet;
    private taskModel: Taskbar;
    private view: UserInterface;

    private theInstance: ProgramController;

    public getInstance(): ProgramController {
        if(this.theInstance == null) {
            this.theInstance = new
ProgramController();
        }
        return this.theInstance;
    }
}
```

```
export class Row implements Dataset {
    private cells: Array<Cell>;

    public delete(): void {
        throw new Error("Method not
implemented.");
    }
    public getCells(): Array<Cell> {
        return this.cells;
    }
}
```

```
export class Spreadsheet {
    private rows: Array<Dataset>;
    private columns: Array<Dataset>;
    private cells: Array<Cell>;

    private theInstance: Spreadsheet;

    public getInstance(): Spreadsheet {
        if(this.theInstance == null) {
            this.theInstance = new Spreadsheet();
        }
        return this.theInstance;
    }
    public getRows(): Array<Dataset> {
        return this.rows;
    }
    public getColumns(): Array<Dataset> {
        return this.columns;
    }
    public getCells(): Array<Cell> {
        return this.cells;
    }
}
```

```
export class StringLiteral implements Literal {
    private value: string;

    constructor(value: string) {
        this.value = value;
    }

    public getValue(): string {
        return this.value;
    }
}
```

```

export class Taskbar {
  private TheInstance: Taskbar;
  private buttons: Array<Button>;

  public getInstance(): Taskbar {
    if(this.TheInstance == null) {
      this.TheInstance = new Taskbar();
    }
    return this.TheInstance;
  }

  public getButtons(): Array<Button> {
    return this.buttons;
  }

  public buttonPress(buttonIndex: number): void {
    {
      throw Error('not implemented');
    }
  }
}

```

```

export class UndoButton implements Button {
  private pastStates: Array<Spreadsheet>;

  public getStates(): Array<Spreadsheet> {
    return this.pastStates;
  }

  press(): void {
    throw Error('not implemented.');
```

```

export class UserInterface {
  private sheet: Spreadsheet;
  private taskbar: Taskbar;
  private theInstance: UserInterface;

  public getInstance(): UserInterface {
    if(this.theInstance == null) {
      this.theInstance = new
UserInterface();
    }
    return this.theInstance;
  }

  public getSpreadsheet(): Spreadsheet {
    return this.sheet;
  }

  public renderInterface(): void {
    throw Error('not implemented');
  }
}

```

```

export abstract class WindowContent {
  private buttons: Array<Button>;
  private components: Array<Component>;

  public update(): void {
    throw Error('not implemented');
  }
}

```

React-Based Statistical Spreadsheet Software

←

→

↺

Q

https://

☰

Filename

Import File

Export File

Undo

Redo

Formula

Statistics

= 3*B2 + C2

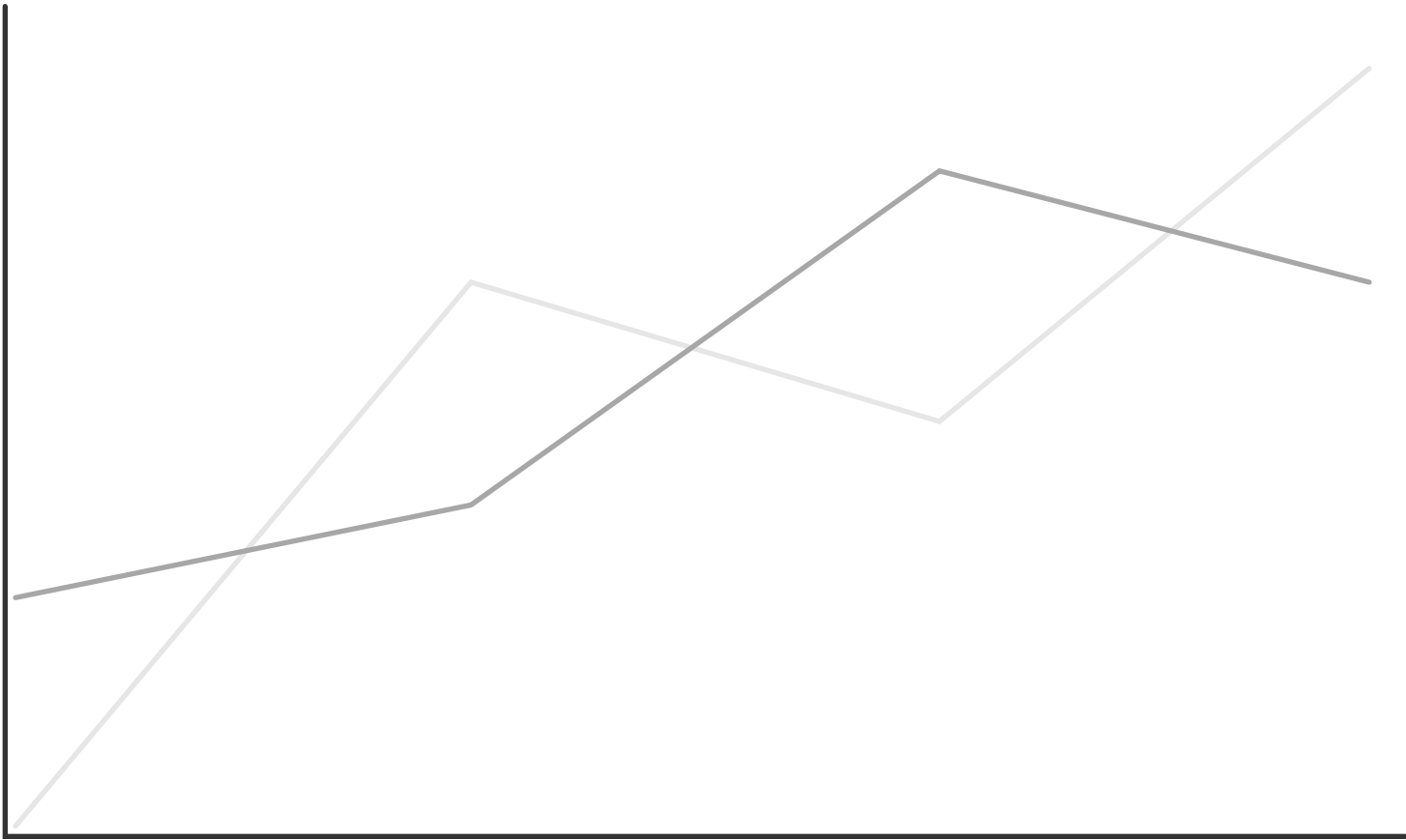
A	B	C		E	F	G	H	I	J	K	L
1	Rory	Keyme	Barplot	m.br	Male	154.188.87.232	= 3*B2 + C2				
2	Beryle	Leavens	Boxplot	k	Female	193.87.19.105					
3	Erick	Whiley	Scatter Plot	e.org	Male	88.143.79.17					
4	Anthea	Kitcat	Line Plot	ption.com	Female	115.62.77.199					
5	Melinde	Bricksey	New T-Test	moncksey@reday.com	Female	191.101.128.63					
6	Thebault	Chelam	New Correlation	tchelam5@jigsy.com	Male	43.2.211.96					
7	Celestyn	Camous	New Linear Model	ccamous6@noaa.gov	Female	22.158.75.246					
8	Tamera	Heathcote		theathcote7@timesonline.co.uk	Female	247.33.238.177					
9	Marje	Gilyatt		mgilyatt8@clickbank.net	Female	218.20.169.48					
10	Dominique	Clayton		dclayton9@timesonline.co.uk	Male	175.135.239.1					
11	Lorraine	Hamberstone		lhamberstonea@nature.com	Female	38.115.152.13					
12	Ferguson	Gatchell		fgatchellb@technorati.com	Male	182.225.16.126					
13	Ealasaid	Demcak		edemcakc@networksolutions.com	Female	119.58.23.156					
14	Geoff	Chinnock		gchinnockd@bandcamp.com	Male	116.129.206.207					
15	Blythe	Gillyatt		bgillyatte@mashable.com	Female	241.128.235.93					
16	Derby	Freckleton		dfreckletonf@sphinn.com	Male	136.123.98.38					
17	Willem	Reinisch		wreinischg@dmoz.org	Male	31.202.159.217					
18	Keefer	Cowup		kcowuph@ustream.tv	Male	70.60.169.241					
19	Bradán	Bruyet		bbruyeti@tripod.com	Male	165.162.35.105					
20	Buckie	McKoy		bmckoyj@epa.gov	Male	81.24.80.29					
21	Giffy	Fuzzard		gfuzzardk@wiley.com	Male	233.240.51.167					
22	Roxanne	Braunstein		rbraunsteinl@noaa.gov	Female	201.111.54.139					
23	Rey	Mc Dermid		rmcdermidm@go.com	Male	33.96.247.100					
24	Benjamin	Hymers		bhymersn@php.net	Male	19.181.128.53					
25	Elmo	Offield		eoffieldo@accuweather.com	Male	245.48.20.89					
26	Gearard	Wayland		gwaylandp@desdev.cn	Bigender	23.170.55.115					



New Graph

Title

Y Axis Name



Var1 ▼

Var2 ▼

X Axis Name



New Linear Model

Parameter	Beta	Standard Error	T Value	p Value
Intercept	0.2	5	0.7	0.86
Param1	0.4	5	0.6	0.75
Param2	2.5	3	4.2	0.003

Add New Variable

- Column A
- Column B
- Column C



Import File

File Name	Size	Kind	Date Added
Folder1	--	Folder	10/4/2023
Folder2	--	Folder	10/6/2023
data1.csv	55KB	CSV	10/15/2023

Path:

Import



Export File

File Name	Size	Kind	Date Added
Folder1	--	Folder	10/4/2023
Folder2	--	Folder	10/6/2023
data1.csv	55KB	CSV	10/15/2023

Filename:

data2.csv

Export