LIST OF REQUIREMENTS

DATA REQUIREMENTS

1. **Spreadsheet Definition:** The application shall define a Spreadsheet as a grid of Cells.
2. **Cell Collection**: A Spreadsheet shall consist of a collection of Cells arranged in rows and columns.
3. **Cell Identification:** Each Cell within the Spreadsheet shall be uniquely identified by a Coordinate.
4. **Coordinate Composition:** A Coordinate shall consist of two parts.
5. **Column Definition:** A column is the first part of a coordinate and is denoted by an uppercase letters (ex.: A, B, AA, etc.).
6. **Row Definition:** A row is the second part of a coordinate and is denoted by an integer positive number.
7. **Range Definition:** A Range shall be a group of Cells in a rectangle shape.
8. **Range Coordinates:** A Range shall be defined by two Coordinates separated by a colon (":")—one for the top-left Cell and one for the bottom-right Cell (ex.: "A1:B3").
9. **Metadata Tracking:** The Spreadsheet shall keep track of the number of rows and columns in use.
10. **Content Types:** The application shall support three primary Types of Content for Cells: TextContent (strings), NumericalContent (numbers), and FormulaContent (computed expressions).
11. **Text Content Value:** Text Content shall have a Value that is a string.
12. **Number Content Value:** Number Content shall have a Value that is a number.
13. **Formula Content Value:** Formula Content shall have a Value that is the result of a calculation.
14. **Default Cell Value:** A cell not edited by the user has an empty string as value.
15. **Formula Start:** A Formula shall start with an equals sign ("=")
16. **Formula Parts:** A Formula shall have Operands and Operators.
17. **Operand Types:** Operands in a Formula shall be Numbers (ex.: 5), Cells (ex.: "A1"), or Functions (ex.: "SUMA(A1:A3)").
18. **Operator Types:** Operators in a Formula shall be "+" for addition, "−" for subtraction, "*" for multiplication, or "/" for division.
19. **Function Arguments:** Functions in a Formula shall have Arguments, which can be Numbers, Cells, Ranges, or other Functions.
20. **Argument Separator:** Arguments in a Function shall be separated by a semicolon (";") (ex.: "SUMA(A1:A3;B1:B3)").
21. **File Format:** Spreadsheets are stored as S2V format when saved.
22. **File Cell Content Representation:** Each line contains the textual representations of the contents of the cells in that row, separated by the semicolon character (;). The S2V file stores only the raw content of cells, not the computed numerical results.

23. **File Row/Column preservation:** The order of cell contents in each line corresponds to the column order (A, B, C, etc.) in the spreadsheet. The number of lines in the file corresponds to the number of rows in the spreadsheet.

FUNCTIONAL REQUIREMENTS

23. **Spreadsheet Management:** The program shall let users create, edit, and delete a Spreadsheet.
24. **Content Value Processing:** The program shall show each Cell's Value: Text as a string, Numbers as a number, and Formulas as their calculated result.
25. **Automatic Dependency Updates:** The program shall update all related Cells when one Cell's Content changes (ex.: if "A1" changes, "=A1+B1" updates).
26. **Load Computation:** After loading a file, the program shall calculate Values for all non-empty Cells right away.
27. **TextContent Number Check:** For Text Content, the program shall: give 0 if empty, a number if it looks like one (ex.: "123"), an error if a number is asked for but it's not a number (ex.: "Hello"), or the string itself otherwise.
28. **NumericalContent Processing:** For Number Content, the program shall: give the number if asked for a number (ex.: 42), a string version if asked for a string (ex.: "42"), or an empty string if the Cell is empty.
29. **Formula Calculation:** Formulas shall calculate Values using other Cells' Values.
30. **Formula Parsing and Evaluation:** The program shall read and calculate Formulas with Operators ("+", "−", "*", "/"), Cell references (ex.: "A1"), Ranges (ex.: "A1:B3"), and nested Functions (ex.: "=SUMA(A1:A3)+MIN(B1:B3)").
31. **Supported Functions:** The program shall include Functions: SUMA to add Values, MIN for the smallest Value, MAX for the largest Value, and AVERAGE for the average.
32. **Complex Expression Handling:** The program shall handle tricky Formulas like "=1 + A1*((SUMA(A2:B5;PROMEDIO(B6:D8);C1;27)/4)+(D6-D8))".
33. **Function Argument Flexibility:** Functions shall accept one or more Arguments separated by semicolons (";"), allowing variable input sizes.
34. **Syntax Error Detection:** The program shall detect and report syntax errors in FormulaContent (ex.: "=A1++B1" is invalid), preventing malformed expressions.
35. **Circular Dependency Prevention:** The program shall stop Formulas that loop (ex.: "A1" = "=B1", "B1" = "=A1") and show an error.
36. **File Saving:** The program shall save a Spreadsheet to a file in the s2v format, enabling data persistence.
37. **File format:** The program must be able to store the Contents of a Spreadsheet in the S2V format.
38. **File compatibility**: The program must also be able to parse the contents of a file whose contents are conformant to the S2V format and create the corresponding Spreadsheet.
39. **Textual Menu Interface:** The program shall provide a textual menu of options (ex.: "1. Edit Cell, 2. Save Spreadsheet, 3. Load Spreadsheet") for users to pick options.
40. **Console Feedback:** The program shall show messages like "Cell A1 updated" or errors on the console.

41. **Error Logging:** The program shall save all errors (ex.: bad Formulas) to a file with times for later review.