Statement of Work

Minimum Viable Product

Minimum set of features required for basic functionality:

- Implementation of a user interface displaying a grid of cells.
- Clearly labeled rows and columns for easy navigation.
- Functionality allowing users to select, edit, and delete the contents of a cell.
    - Support for various data types (texts, numbers, formulas).
    - Support for basic mathematical functions: addition, subtraction, multiplication, division, and summation across rows and columns.
    - Formulas are recalculated when their dependency cells are updated.
- Ability to create a new spreadsheet.
- Ability to save a sheet on the server.
- Ability to delete a sheet on the server.
- Ability to see which sheets are available on the server.
- Ability to choose a sheet from the server and get the most recent version of it.
- Mutually recursive formulas do not crash the program.
- Copy/Cut and paste for single cells.
- Support for sheets up to 100 by 100 (10000 cells)

Desirables

More complex set of features for enhancing project functionality (*Don't include features we are uncertain about to avoid losing points*):

- Confirmation to the user when updating the server.
- Basic cell formatting options: font styles, text alignment, background colors, and borders.
- Error handling mechanisms for invalid inputs, formula errors, and other issues.
- Resizable rows and columns.
- Select the whole row/column.
- Select the whole region.
- Copy/Cut and paste for regions.
- Shift-click and control-click to add or remove from the selected region.
- Add a new row above/below the current row.
- Add a new column right/left of the current column.
- Delete row/column.

- See which sheets are being edited by another user.
- Auto-save option.
- Support for larger sheets up to 1 million cells

Bonus Features

Additional features for extra credit:

- Highlighting the entire row when being accessed (Google Sheets doesn't do this).
- Smart copying of formulas where pasting updates relative cell numbers.
- Dragging a cell to copy/extend (for example dragging on the number 1 to get numbers in order).
- Support for multiple file formats: CSV, JSON, etc.
- Advanced formatting options: conditional formatting, cell merging, etc.
- Importing and exporting data from external sources: Excel files, Google Sheets, and other databases.
- Version history for documents and the ability to roll back to a previous version.
- Diff and conflict system so multiple people can check out the same sheet.
- Support for pulling data from external databases in formulas.
- Charting and graphing capabilities to visualize data.
- Different number formatting options (currency/time/number of decimal places).
- Support for collaborative editing so users can concurrently view and edit. (exceeds MVP sequential collaboration requirement).

TA Name: Shubh
TA GitHub: shubhdesai1611
TA Email: desai.shu@northeastern.edu
Breakout Room: 4

Components

- GUI (JavaFX)
    - Sheet Display
        - Grid layout for displaying cell data with resizable columns and rows.
        - Highlight active cell and selection range for clarity.
        - Real-time updates to cell values and formulas.
    - Menu for Choosing Sheet
        - Dropdown or sidebar menu to list available sheets.
        - Options to create new sheets or delete existing ones.
        - Functionality to rename sheets directly from the menu.
    - Menus for Choosing Formatting Options
        - Toolbar with buttons for font style, size, bold, italic, underline.
        - Color picker for text and background color.
        - Tools for cell alignment (left, center, right), text wrapping, and borders.
- Client
    - Data Handling
        - Manage user sessions and interactions with the spreadsheet.
        - Send requests to the server for saving, retrieving, and updating sheets.
    - User Authentication
        - Implement user login, registration, and session management.
        - Ensure secure transmission of credentials and use tokens for session validation.
    - Error Handling
        - Provide user-friendly error messages and logging for debugging.
        - Implement retry mechanisms for failed requests due to network issues.
- Server

- API Endpoints
    - ‘POST /sheets’ to create a new sheet.
    - ‘GET /sheets’ to retrieve a list of sheets.
    - ‘PUT /sheets/{id}’ to update an existing sheet.
    - ‘DELETE /sheets/{id}’ to remove a sheet.
    - Additional endpoints for specific actions like updating cell values or batch operations.
- Data Storage
    - Use a database to store sheet data, user information, and session logs.
- Business Logic
    - Implement the core logic for how users interact with sheets.
    - Ensure consistency and integrity of data through transactional controls and validations.
    - Optimization strategies for handling large sheets and concurrent users.