**SheetCell Application – README**

**1. Overview of the System**

The spreadsheet management system allows users to create and update spreadsheets, track changes through versioning, and manage cell dependencies. It uses object serialization to save and load the entire state of the system, making it easy to persist and restore spreadsheets.

**2. Explanation of Main Classes**

- ExpressionFactoryImpl: factory for creating expressions and validating their arguments. It ensures that expressions are properly instantiated and that their arguments meet the required criteria.

-ConverterUtil: provides methods to convert internal data structures to DTOs, facilitating data transfer and representation.

-Logic: provides functionality for managing spreadsheet data, including adding new sheets, updating cell values, and accessing different versions of sheets.

-VersionManager: manages a list of sheet versions, allowing you to add new versions, retrieve specific versions, and get the latest version.

-CellImpl: represent and manage the state and behavior of a cell in a spreadsheet.

-EffectiveValueImpl: encapsulates the value of a cell, including its type (numeric or string) and its evaluation based on expressions.

-SheetImpl: manages the state of a spreadsheet, including its cells, dependencies, and versioning. Edge - dependency between cells.  
important functions:   
onCellUpdated: Handles the update of a cell's value and recalculates dependencies.  
updateCells: Updates cells that are dependent on the given cell and recalculates their values.  
orderCellsForCalculation: Orders cells topologically based on their dependencies to avoid circular dependencies and ensure proper evaluation order.

-UIManager: provides a user interface for interacting with the spreadsheet application. It handles user input and output, menu operations, and file management.  
  
It's important to note that the calculation of ref is done dynamically in the system. Therefore, messages such as NaN and undefined will only appear after the calculation of values  
These messages will be displayed if the arguments result in a failure during calculation.

**3. Bonus Implementation** System Save and Load Design Choices Saving and Loading: I decided to use object serialization for saving and loading the system state.

**4. Submitter Details**

* **Name:** Noy Zion
* **ID Number:** 212198766
* **Email:** noyzion3@gmail.com