

## I. Base use case [Gray Buckley]

The server will listen for TCP connections on port 2112. The client can connect with any valid message as described below.

Messages start with a special character from the set {`$`,`?`,`&`,`!`}. Messages from the clients to the server start with a dollar sign or a question mark, and messages from the server to the clients start with an ampersand or an exclamation mark. A dollar sign indicates a change request, a question mark indicates a query, an ampersand indicates a description, and an exclamation mark indicates an error message. All messages end in a semicolon, after which another message may or may not begin.

A message consists of the special character indicating the message type, followed by one or more comma separated sections, followed by a semicolon.

The general form for sections are `A A:B` and `A:B:C` where `A` is a spreadsheet name, `B` is a cell name, and `C` is the contents of a cell.

A section of a change request may indicate just a spreadsheet name to request a blank spreadsheet with that name be created, or may request that the contents of a cell change by indicating the spreadsheet where the change is requested, the cell where the change is requested, and the contents the cell is to be changed to.

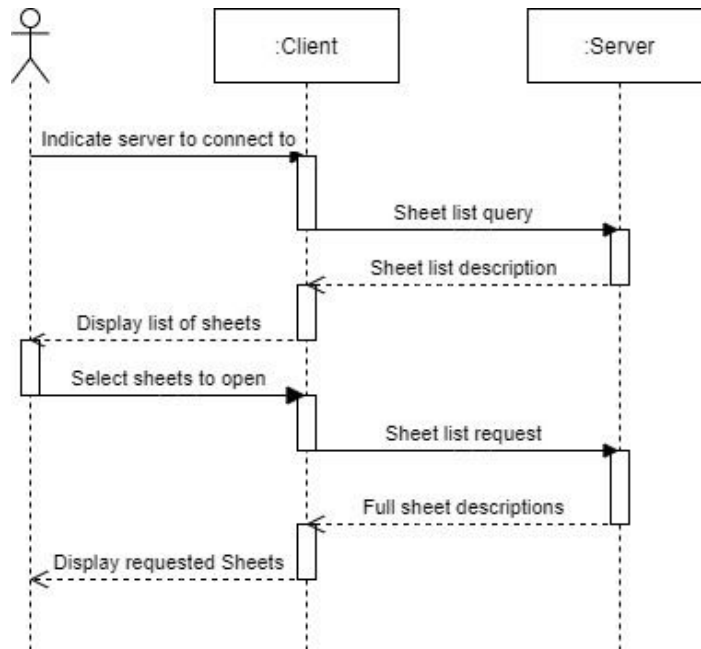
A query may be of the special form `??`; to query a list of spreadsheets on the server, or it may consist of sections as in the general form. A section may just specify a spreadsheet to query a full description of that sheet, or it may specify a spreadsheet and a cell to query a description of just that cell in that sheet.

A section of a description may indicate the spreadsheet of the described cell, the described cell, and the contents of the described cell. A section of a description may also indicate just a spreadsheet name to indicate its existence.

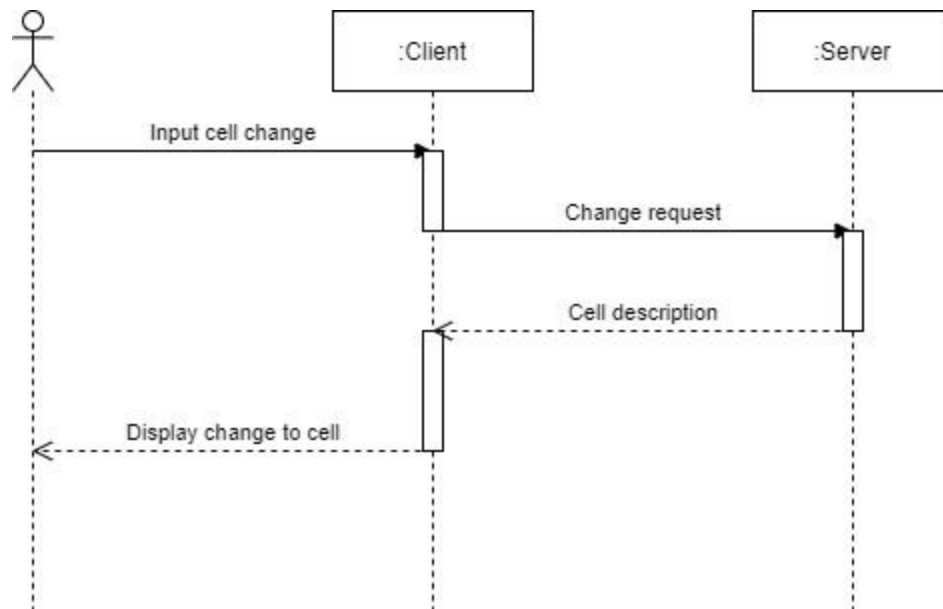
An error message consists of comma separated sections ending in a semicolon like other messages but each section is a plaintext description of an error rather than the name of a spreadsheet.

If a character from the set {`$` `?` `&` `!` `,` `;` `:` `~`} is used in a spreadsheet name, cell name, cell contents, or error message, it can be used with the backslash as an escape character (ex. `\$` for `$`). The backslash itself can be used as `\\`.

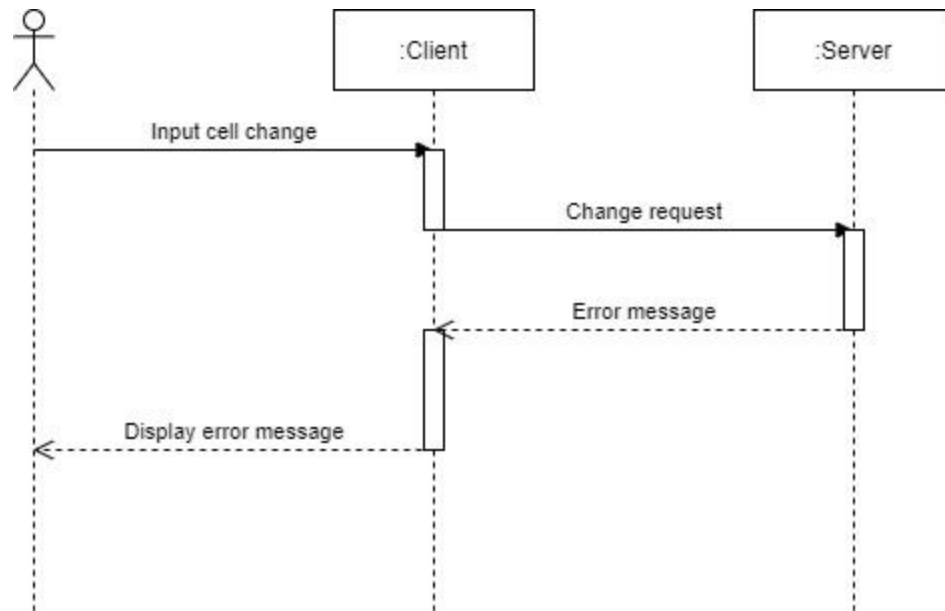
Likely scenarios for these combinations of messages include:



*A client connects to a server. The client queries a list of spreadsheets. The server sends a description of what spreadsheets are in the server. The user selects one or more spreadsheets. The client sends a query for all the selected sheets. The server sends descriptions of all the selected sheets.*



*A client sends a compatible change request to the server. The server determines that the change is valid, and sends back a description of the changed cell to indicate that the change was accepted.*



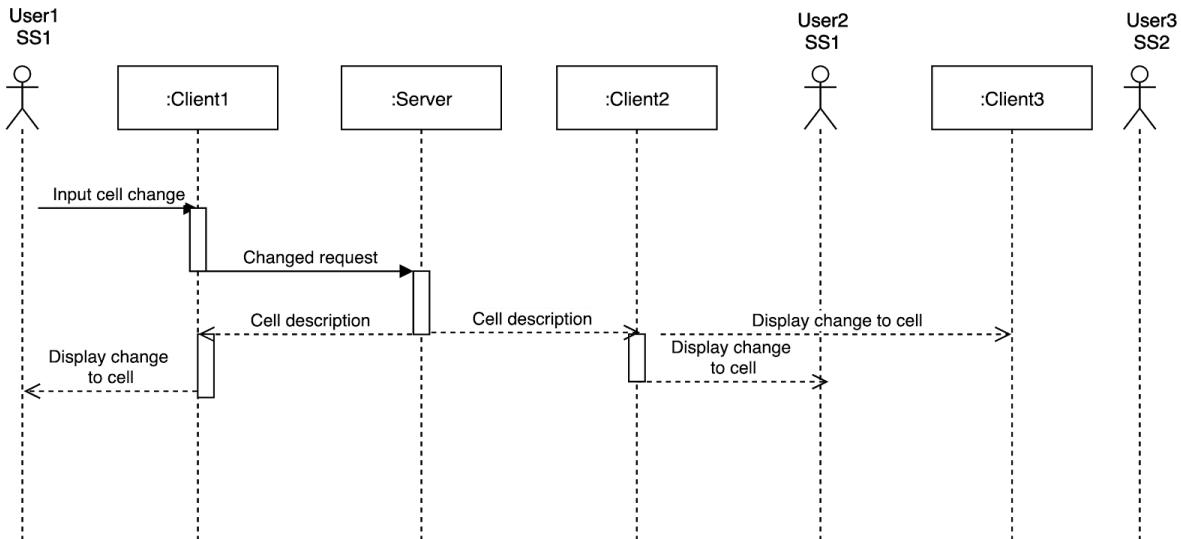
*A client sends an incompatible change request to the server. The server determines that the change is invalid, and sends back an error message describing why the change was denied.*

## II. Multiple users [Jesse Sanchez]

The server will listen for multiple connection requests to allow multiple clients to access the spreadsheets saved on the server. This will allow multiple users to work on the same spreadsheet together or work on different spreadsheets separately. The clients will only send messages to the server. A client cannot send a message to another client.

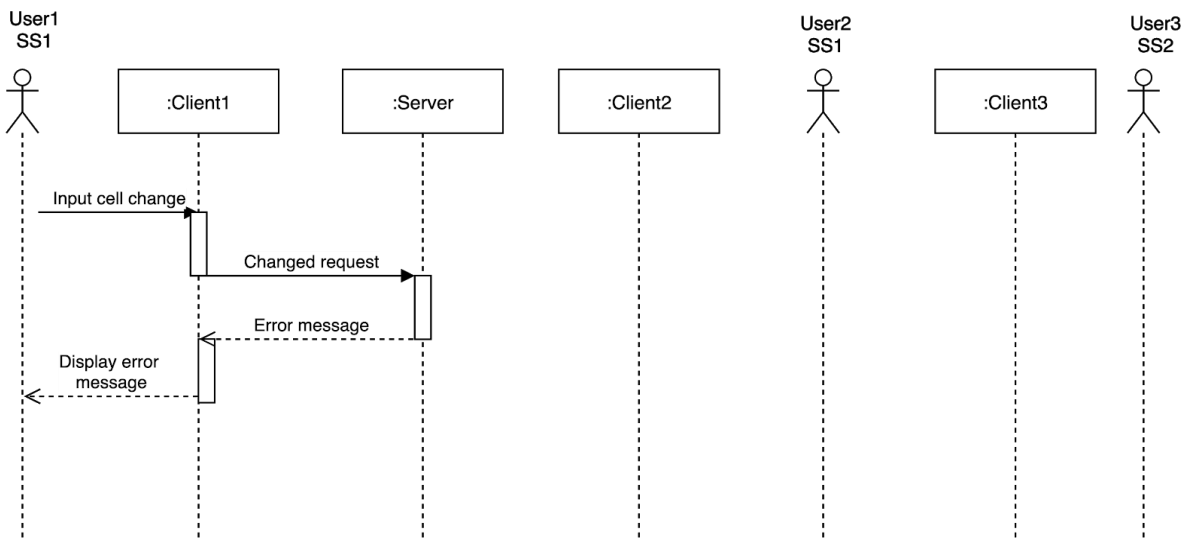
When a user successfully connects to the server, the server will respond to the client as described in the **base use case**. When one or more clients connect to the server, each client may or may not receive a response whenever the server receives a message from any client. Changes made by clients may or may not make changes to a spreadsheet. Successful changes will be made in real time and these changes will be sent as messages by the server to all other clients connected to the server. In the case of an unsuccessful change, the server will only respond with an error message to the client of the user that made the error.

If the server receives a message by a client to make a change in a spreadsheet and the change is **successful**, the server will then send a message to all clients. The clients will determine if the message is relevant to the user.



*In this example, User1 and User2 are both editing the same spreadsheet called SS1. User3 is editing a spreadsheet called SS2. User1 submitted a cell change to the server. The server responded by sending a message of the cell description to all three clients. Client1 and Client2 display the cell change for user1 and user2 respectively. Client3 chose not to display anything for user3 as the spreadsheet is not the same.*

If the server receives a message by a client to make a change in a spreadsheet and the change is **unsuccessful**, the server will then send a message describing the error to the client of the user that made the invalid change.

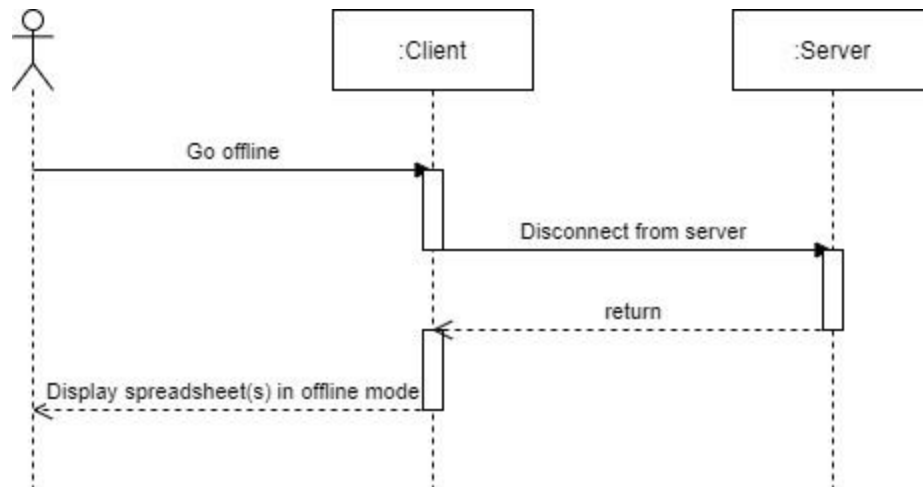


*This shows how User1 attempted to make a cell change on spreadsheet SS1 that resulted in an error. The server responded to Client1 with an appropriate error message which was displayed to user1. Client2 and client3 do not receive the error message.*

When a user decides to log off, the server will stop sending messages to the user's client. The user's client will decide what to do after logging off.

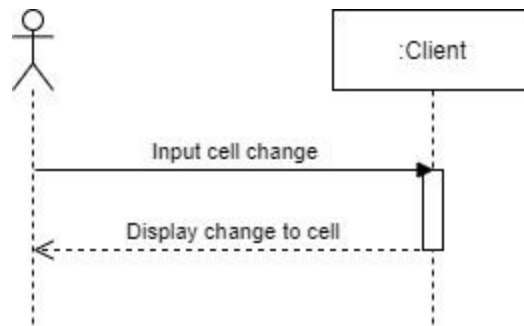
### III. Offline use (Jose Mattam)

When using a spreadsheet program, users are given the option of using offline mode. This will disconnect the client from the server and let the user work on all open spreadsheets without a network connection. All changes made to a spreadsheet in offline mode will be lost if the client is not connected to a server before the spreadsheet is closed. Implementation dictates whether or not errors are checked when in offline mode.



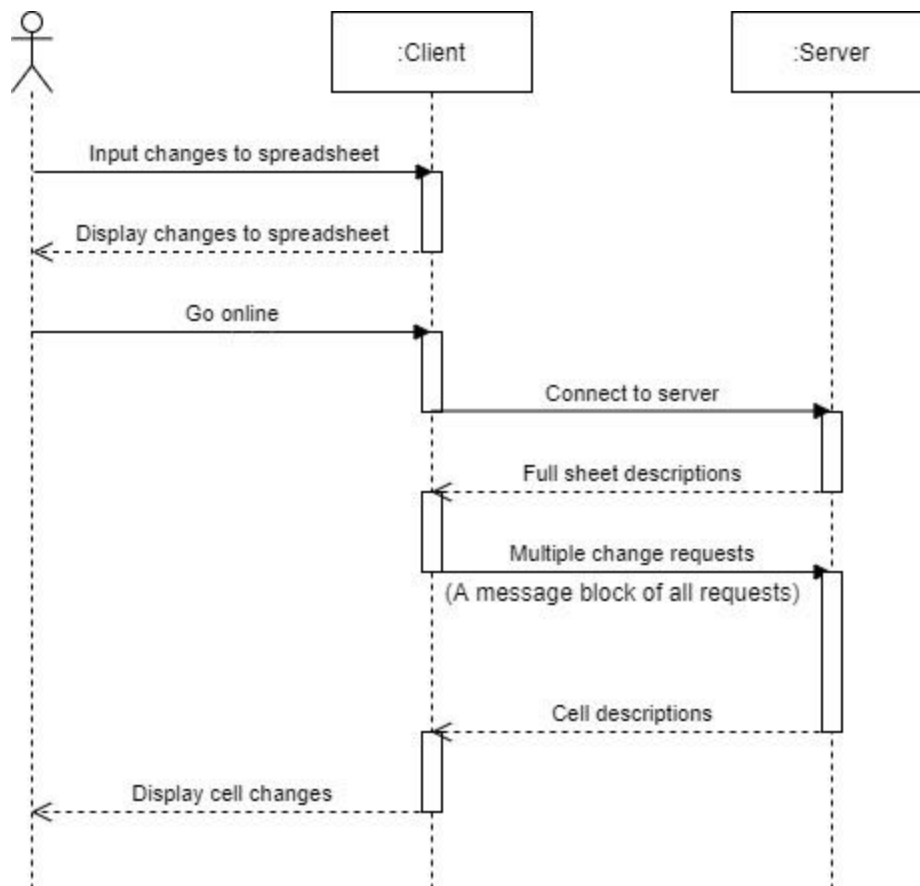
*In this example, the user is at first, connected to a server and using a spreadsheet. The user then chooses to use the offline mode. The client gives the user the last saved changes of the spreadsheet(s), now usable without a network connection.*

Changes to a cell of an offline spreadsheet don't go to the server. They are automatically displayed to the user.



*The user is in offline mode and makes changes to the spreadsheet.*

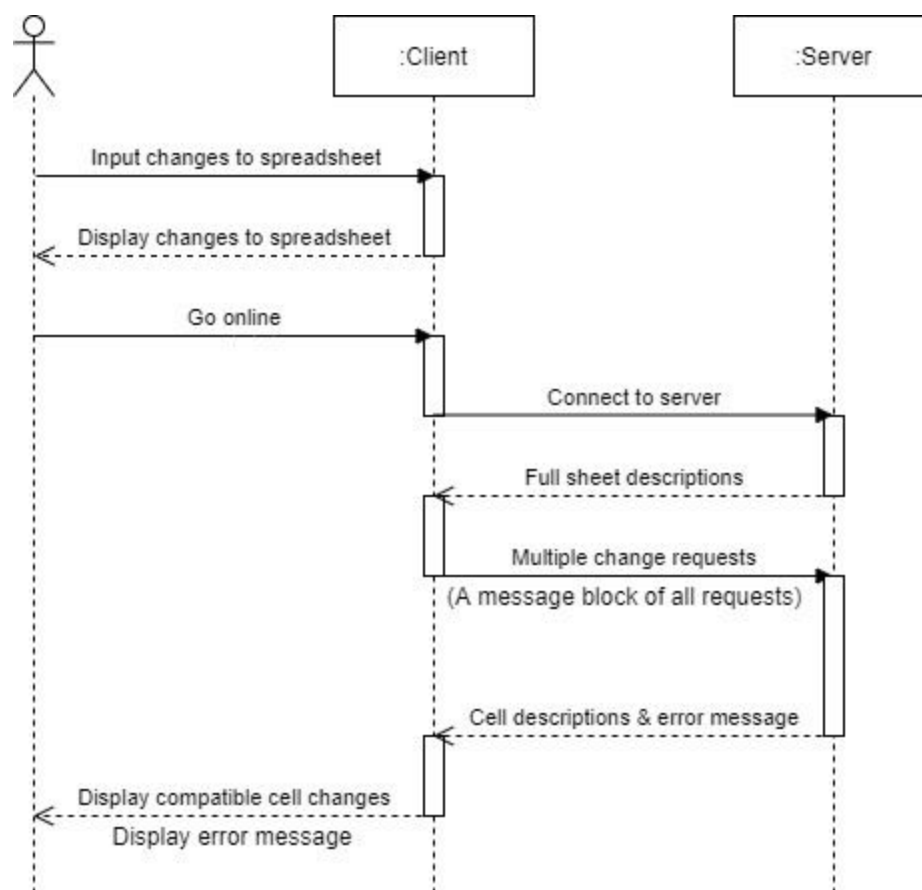
When turning off offline mode, all required changes made in the offline mode spreadsheet will be sent from the client as change requests to the server. If no errors occur when making those changes, the server sends back descriptions of the changed cells to indicate that the changes were accepted.



*The user is in offline mode and reconnects to the server by turning off offline mode.*

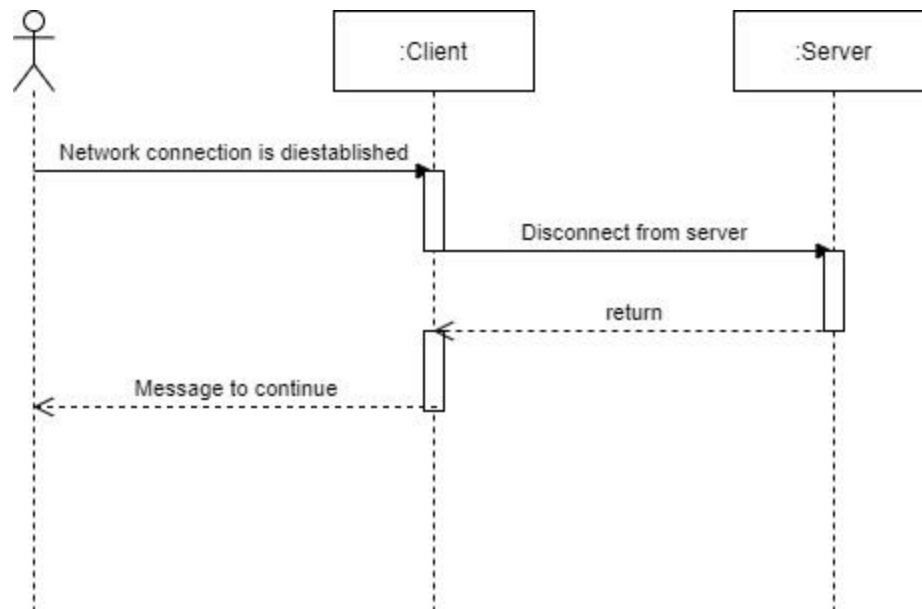
*The client then reconnects to the server and retrieves the spreadsheet's description. This is then passed back to the client before any of the offline changes are made and displayed.*

When reconnecting to the server once offline mode is turned off, the server will search for any errors that were created in the merge of the offline and online spreadsheets. This includes formula errors and circular dependency errors. If any of the change requests cause such an error, they are seen as invalid. All valid change requests are executed and displayed without modification. All invalid change requests are grouped together into a single error message which is sent back to the client and displayed to the user with explanations of why the errors occurred.



*In the case of any errors when merging the offline and online spreadsheets.*

When a user disconnects from the network used, the client disconnects from the server and displays a message to the user giving them the option to either go offline or close the program.



*A user who is using a spreadsheet suddenly disconnects from their network. The client disconnects from the server and displays a message for the user to continue using the spreadsheet.*

#### IV. Undo/Revert (Brandon Walters)

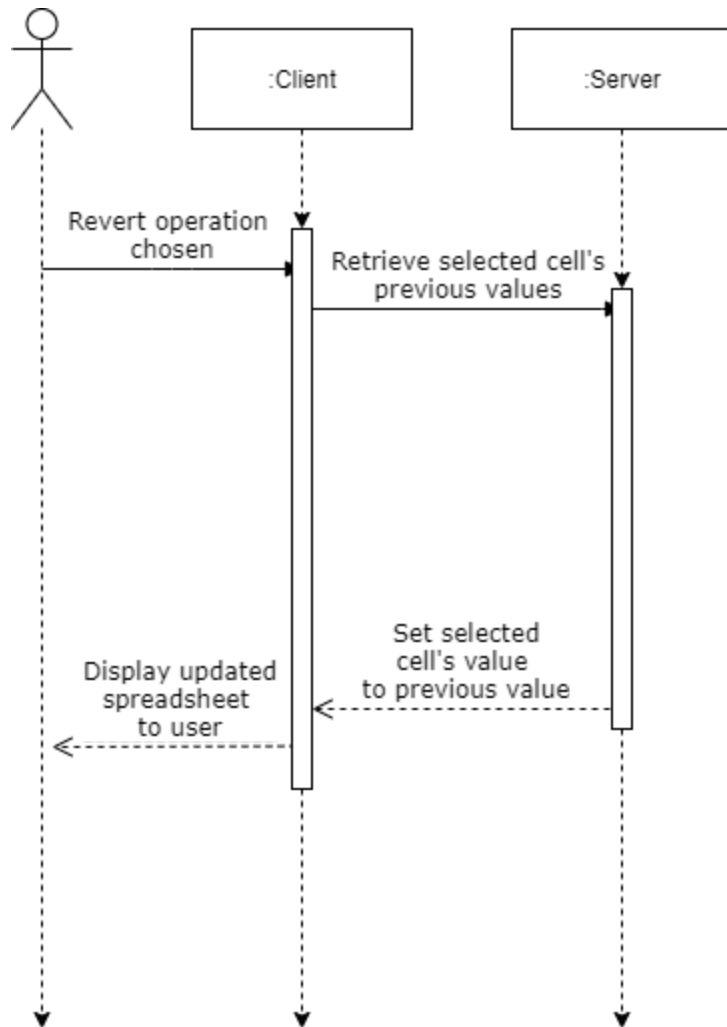
Users will be allowed to revert changes to the spreadsheet and undo the last cell edited on the spreadsheet to a previous state. These changes will act the same as normal spreadsheet changes, and work may possibly be overwritten, including that of other clients working on the same spreadsheet. Multiple users may undo and revert the same spreadsheet simultaneously.

Undos and reverts use a special variant of change request sections, which may be mixed in with other change request sections in a change request message. A section indicating an undo consists of a tilde followed by the name of the spreadsheet the undo is to be performed on. [~A] A section indicating a revert consists of the name of the spreadsheet containing the target cell, a colon, a tilde, and then the name of the target cell. [A:~B]

A revert operation only affects the selected cell. If the user selects the revert on a selected cell, the cell's value will be stepped back to the last saved value, which is stored in the server. A user can repeatedly revert a cell, to eventually step all the way back to the

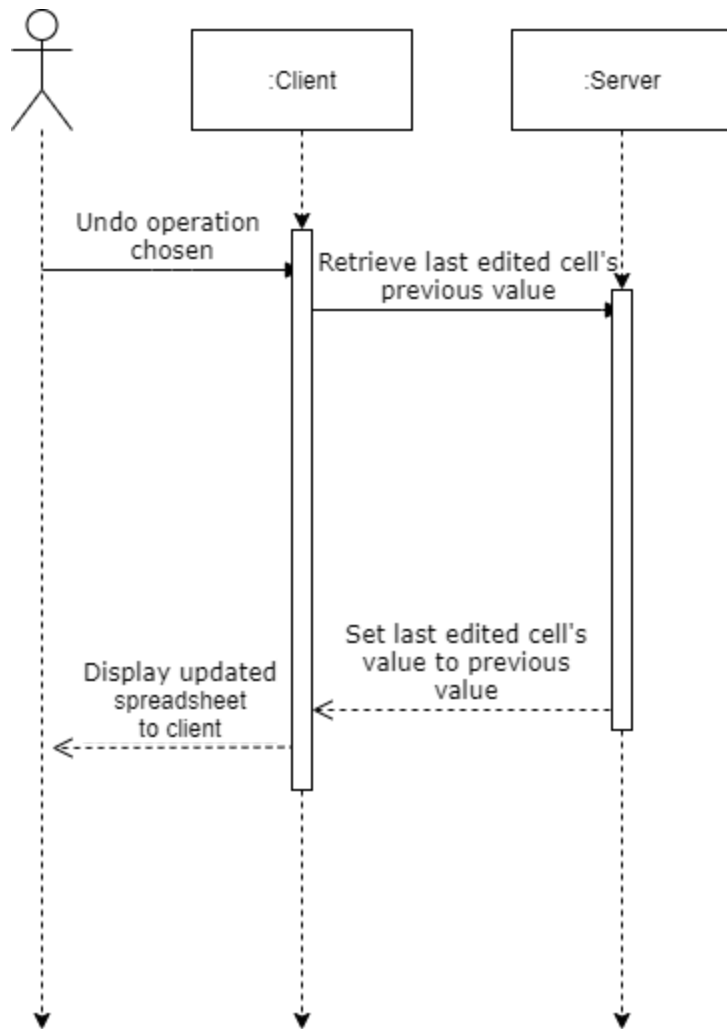


original state of the cell upon the spreadsheet's creation.



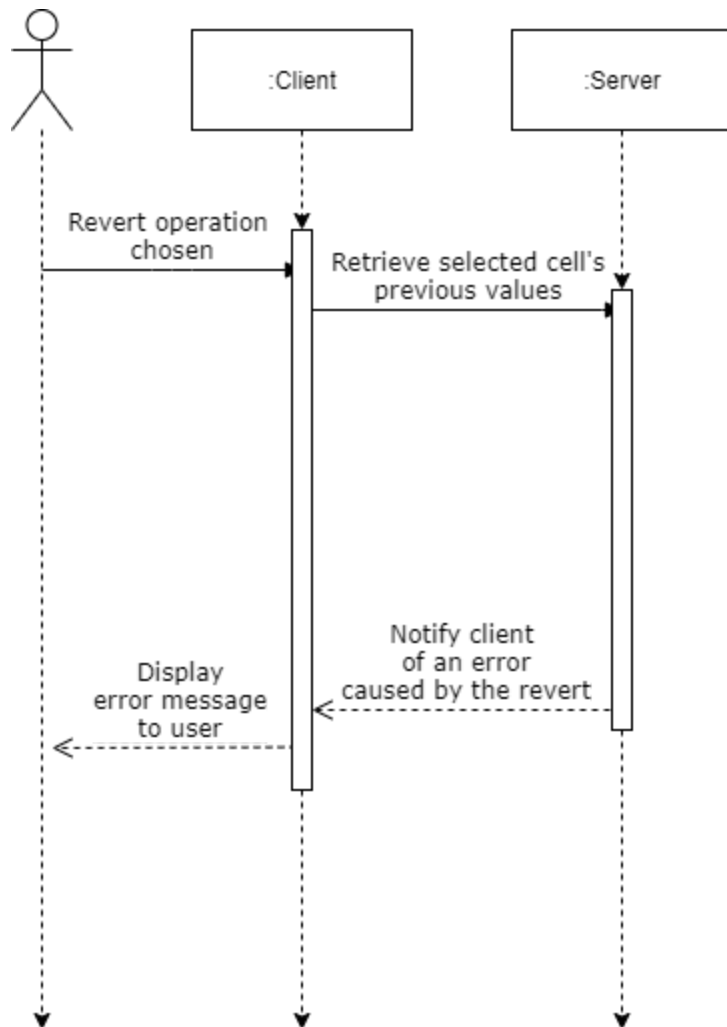
*A typical revert operation.*

An undo operation affects the last edited cell in the sheet, returning it to its prior value. A user can repeatedly undo a spreadsheet, to eventually step all the way back to the original state of the spreadsheet upon its creation.



*A typical undo operation.*

If an undo/revert operation causes an error or circular dependency in the spreadsheet, the server will notify the client of the error or errors that have been caused. The client will then display a message to the user that notifies them of the error or errors caused.

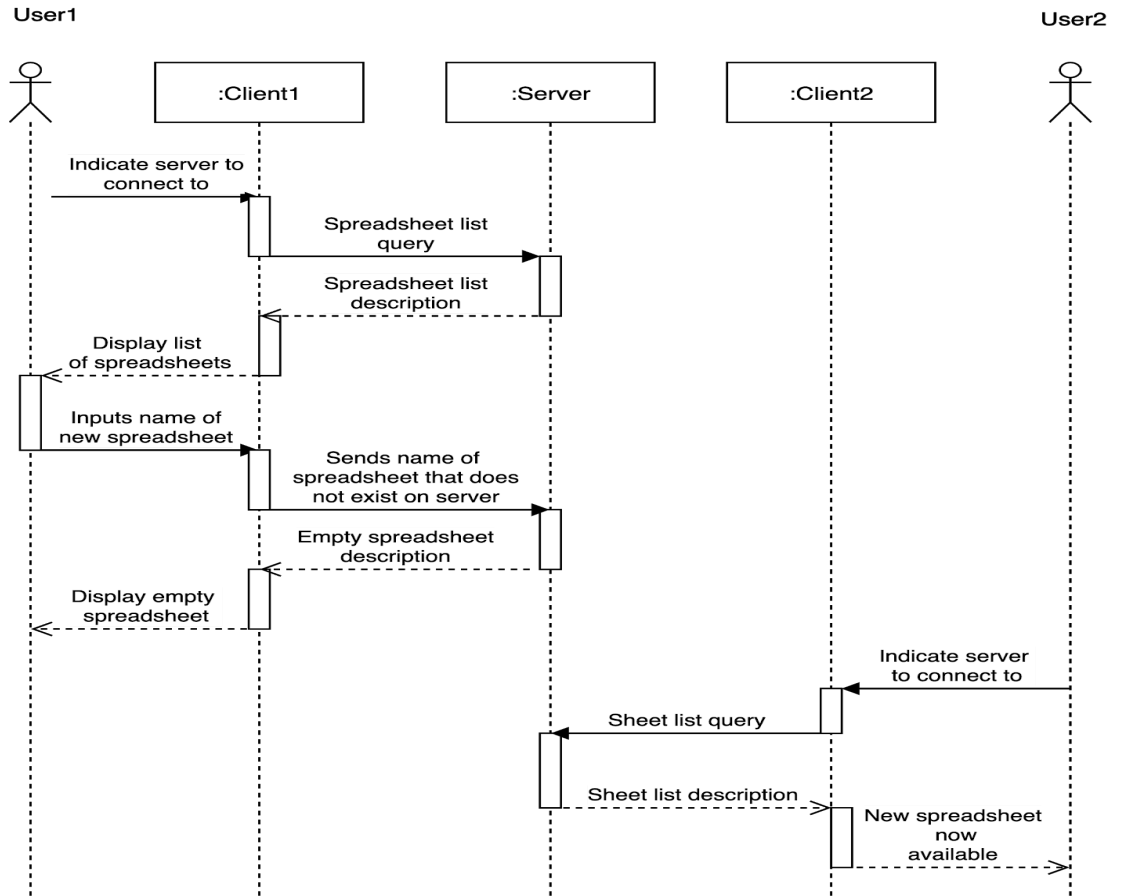


*A revert operation that causes an error of some kind*

In offline mode, the undo and revert functions will be limited. The undo and revert operations will only work to the beginning point of the spreadsheet's history in offline mode. If a spreadsheet was created and edited before offline mode was selected, the undo/revert functions will not allow the user to go all the way back to the spreadsheet's beginning, only to the point when offline mode was selected. Depending on the implementation of offline mode, which may not check for errors, undo/revert operations can be allowed to create errors, such as circular dependencies. See the **Offline Use** section for more details.

## V. Creating new sheets [Jesse Sanchez]

Users will be able to create a new empty spreadsheet using the message format described under base use case. The new spreadsheet can be shared with all users.



*If the user sends the name of a spreadsheet that does not exist on the server, the server will return the description of a new blank spreadsheet. The newly created spreadsheet will then be available for all clients that query for the list of spreadsheets afterward.*

This table surmises the different message types and applicable formats

Message Type	Associated Characters	Sheet	Sheet and Cell	Sheet, Cell, and Contents	Special format
Change Request	\$	Y	N	Y	N
Change Request (Undo/Revert)	~, ~	Y	Y	N	N
Query	?	Y	Y	N	Y
Description	&	Y	N	Y	N
Error Message	!	N	N	N	Y

Example messages:

\$SomeSheet:A5:23;

Changes cell A5 in spreadsheet SomeSheet to 23.

\$SomeSheet:B4:-5,SomeSheet:C4:B4+10;

Changes cell B4 in SomeSheet to -5, and C4 to B4 + 10.

\$NewSheet;

Creates a blank sheet called NewSheet.

\$~SomeSheet;

Undoes the last change within SomeSheet.

\$SomeSheet:~A3;

Reverts the last change to A3 in SomeSheet.

??;

Queries a description listing the sheets on the server.

?SomeSheet;

Queries a full description of all the nonblank cells in SomeSheet.

?SomeSheet:B3;

Queries a description of cell B3 in SomeSheet.

&SomeSheet,AnotherSheet;

Description indicating that the server contains the sheets SomeSheet and AnotherSheet.

&SomeSheet:A6:15;

Description indicating that cell A6 in SomeSheet contains the value 15.

!"SomeSheet:B5:A4-1" causes a circular dependency;

An error message with the text

*"SomeSheet:B5:A4-1" causes a circular dependency.*