

Statement of Work - Team 8

Before starting to code, each team should write a SOW that describes what they expect to implement in terms of MVP (minimal viable product), desirable features and additional features. You will be graded based on meeting that SOW. Use best-practices to document the SOW.

Requirements:

- Client:
 - Sheets that can compute values of cell based on formulas
 - Mediate interactions with UI and the server
- GUI
 - Ability to display and edit sheets
- Documentation on how to use it

Extra Features:

- Login Feature
- Theme Picker
- Independent Save History
- Copy sheets

Endpoints:

- register() causes the server to create a publisher with the client name. No value is returned.
- getPublishers() returns a list of argument objects with the publisher field set to all registered publishers.
- getSheets(arg) takes an argument object with field publisher set to the name of a publisher and returns a list of argument objects with the publisher and sheet fields set to all sheet names for the given publisher.
- createSheet(arg) takes an argument object with fields publisher and sheet set to the name of the client and the name of a sheet to create. No value is returned.
- deleteSheet(arg) takes an argument object with fields publisher and sheet set to the name of the client and the name of a sheet to delete. No value is returned.
- getUpdatesForSubscription(arg) takes an argument object with fields publisher, sheet and id set to the name of a publisher, a sheet, and an id. It returns an argument object with the payload set to all updates that occurred after id, and the id field set to the last id for those updates. The sheet is owned by a publisher different from the client. An empty payload is returned if no updates occurred after the given id.
- getUpdatesForPublished(arg) takes an argument object with fields publisher, sheet and id set to the name of a publisher, a sheet, and an id. It returns an argument object with the payload set to all the requests for updates that occurred after id, and the id field set to the last id for those requests for updates. The sheet is owned by the client. An empty payload is returned if no updates occurred after the given id.

- `updatePublished(arg)` takes an argument object with fields `publisher`, `sheet` and `payload` set to the name of a publisher, a sheet, and updates for that sheet. No value is returned. The sheet is owned by the client.
- `updateSubscription(arg)` takes an argument object with fields `publisher`, `sheet` and `payload` set to the name of a publisher, a sheet, and requests for updates for that sheet. No value is returned. The sheet is owned by a publisher different from the client.

User Stories

1. As a business owner, I want to save a spreadsheet update, so that I can access the spreadsheet data at a later time.

Acceptance Criteria (conditions of satisfaction)

1. **Last Save:** When pulling updates from the server about a particular spreadsheet, the return value is the spreadsheet that was last saved by a user and was approved by the publisher.
 2. **Feedback:** When the data has been saved a visual indication should show that the save is successful / in progress.
 3. **Error Handling:** In case the save fails, the user should receive an error message explaining why the save was unsuccessful.
-

2. As a fashion designer, I want to choose my GUI's theme, so that I have a nice color scheme to organize brands and outfits.

Acceptance Criteria (conditions of satisfaction)

1. **Feedback:** When a theme has been selected, there should be a visual indication that shows the selection was successful.
 2. **Persistency:** Whenever the user closes the application, the last selected theme should still be displayed.
-

3. As a manager, I want to see my worker's independent save histories, so that I know who is contributing and doing their job.

Acceptance Criteria (conditions of satisfaction)

1. **Correctness:** When checking the list of updates on a spreadsheet, each save history should display the correct user editor and the correct edits that were made.

2. **Persistence:** Whenever the user closes the application, all of the user save histories should be stored and accessible on another open.
-

4. As a database administrator, I require my users to register a unique name and to follow a set of instructions when creating their passwords, so that the application can avoid the same username errors and users have their data protected.

Acceptance Criteria (conditions of satisfaction)

1. **No Same Names:** Users must choose a unique username when registering.
 2. **Data Protection:** Users must follow a set of guidelines that enforce strong passwords when registering.
 3. **Error Handling:** If a user chooses a username that has already been taken by another user, an error message will be displayed.
 4. **Feedback:** On a successful registration, a visual indication will be displayed.
 5. **Login:** To login, users must enter their unique username and unique password to gain access to their spreadsheets.
-

5. As a project manager, I want to be able to check and approve updates to my spreadsheets, so that I can ensure the updates are not malicious.

Acceptance Criteria (conditions of satisfaction)

1. **All Updates:** The publisher should be able to view all requested updates since the last update to the spreadsheet.
 2. **Manual Selection:** The publisher should be able to select which updates they approve and reject.
 3. **Error Handling:** If an error occurs with saving the spreadsheet, an error message will be displayed and the attempted update will not through (latest successful update will remain).
 4. **Visual Feedback:** On successful spreadsheet update, a visual indication will be displayed.
-

6. As a student, I want to copy spreadsheets that my instructor has posted, so that I can become the owner of my own copy and edit it further.

Acceptance Criteria (conditions of satisfaction)

1. **History Not Copied:** The copy will only copy over the sheet data, not the history
2. **Different Copy Name:** The copy will be named "Copy of x"

3. New Copy Ownership: The user is the owner of the new copy

7. As a student, I want to create and delete spreadsheets, so that I can organize my school-work research properly.

Acceptance Criteria (conditions of satisfaction)

1. **Creation:** When creating a new sheet, the user has the option to name it and edit. They also become the publisher of the sheet, which is shown by a visual indication.
 2. **Deletion:** When attempting to delete a sheet, the user must be the publisher. There will be a visual indication that asks for confirmation before deleting the sheet. All data will be erased.
-

8. As a game designer, I want to use functions and operations on my spreadsheet, so that my mathematical work is more efficient.

Acceptance Criteria (conditions of satisfaction)

1. **Correctness:** When using a function on a column, the operations (such as '+', '-') should perform the correct work.
2. **Error Handling:** When using operations, in the case of mismatched or incorrect typing, the user will be presented a small error message to indicate that.