DOCUTAGS PROPOSAL

Project Description:

Docutags is a tag-based file organizer app for students and writers that helps users quickly find documents or content within documents that match their given search and filter criteria.

Competitive Analysis:

Competitor	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Google Docs	Collects files together, displays name of file and timestamp	Sort options: by last modified/last opened/Title (A-Z)	Can select chunks of text and write comments, which can be viewed on the side of the screen on the proper page.	Search bar: Finds files with the matching string in title. Not caps sensitive	Pages are presented in a scrolling fashion, displays numbering of current page out of total pages
TagSpaces	Can add multiple tags with a custom color to imported files. Browse tags in the tag library. Can be filtered for items that have multiple tags	Take notes in plaintext and html. Can highlight text, bold, italicize, and create bulleted lists.	Search and filter will display a count of matching files.	Tags can be grouped into "tag groups". Ex: tag group called "family", which collects the tags of individual family members. Tag groups can have a custom color too.	Files can be collected into folders. Browsing file library offers list, grid, and folder tree options.

Tag and annotate every kind of file or folder

✔ Organize your photos, documents, music, recipes or invoices with tags in the same way on Windows, mac OS, Linux or Android.

✔ TagSpaces features basic file management operations, so it can be used as simple file manager.

✔ You can assign custom color to every folder and tag, which makes the visual search an easy step.

✔ You can add a custom thumbnail and text description to every file or folder.

✔ In contrary to the most file tagging applications, TagSpaces does not persists the tags in a centralized way. As a consequence, the added meta information is not vendor locked.
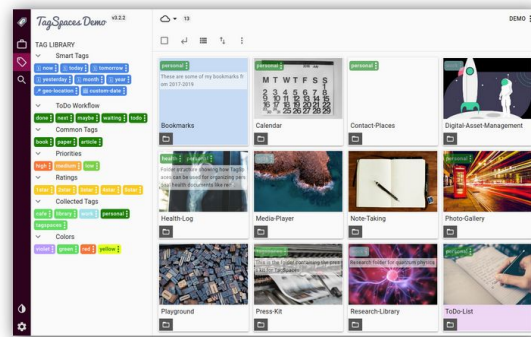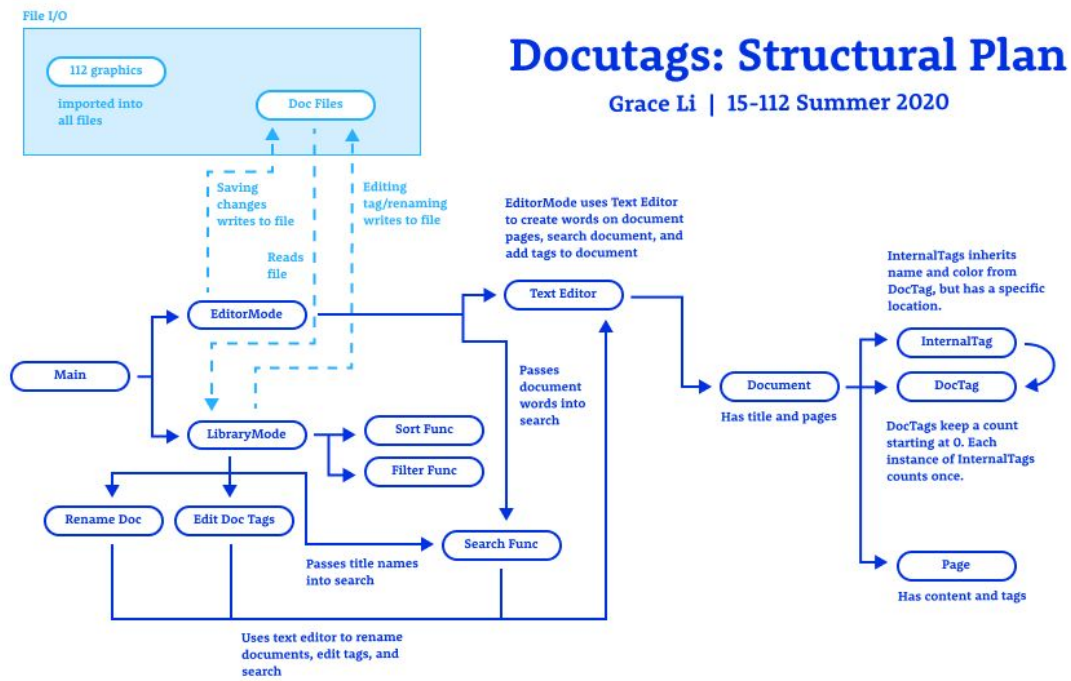
Image: TagSpaces

Unlike its competitors, Docutags is built specifically for note takers and writers for managing text files. Whereas Google Docs is known for its text editor functionality and sharing features, and TagSpaces is designed for all-purpose tagging, Docutags offers a noise-free way of organizing and re-visiting your writing.

Special features:

Feature	Description
Doc tags	General tag attached to the whole document. Not attached to specific internal content. Can be used to sort and filter your documents.
Internal tags	Tag attached to text content. Each instance of an internal tag will be counted and displayed on the document, and internal tags will automatically be added to the list of doc tags.
Filter by multiple tags	Rather than displaying items that have both of the listed tags, docutags will allow users to view a set of files that has any one of the user's chosen tags. For example, if a user wants to review their notes under the tag "recursion" as well as "graphics", they will be able to quickly find all the notes that contain either the tag "recursion" or "graphics"

Structural Plan:

Docutags: Structural Plan

Grace Li | 15-112 Summer 2020

File I/O

112 graphics imported into all files

Doc Files

Saving changes writes to file

Editing tag/renaming writes to file

Reads file

EditorMode uses Text Editor to create words on document pages, search document, and add tags to document

InternalTags inherits name and color from DocTag, but has a specific location.

Main

EditorMode

Text Editor

InternalTag

DocTag

Document

Has title and pages

DocTags keep a count starting at 0. Each instance of InternalTags counts once.

Passes document words into search

LibraryMode

Sort Func

Filter Func

Rename Doc

Edit Doc Tags

Search Func

Page

Has content and tags

Passes title names into search

Uses text editor to rename documents, edit tags, and search

Algorithmic Plan:

Sorting:

```
#########################################
#Sort
#########################################
#from operator import itemgetter, attrgetter

#sort keys: title, time, tag
#mutable/destructive sort which changes the indexes of the items in the list

#def sortTime (sorts by time)
    #documents.sort(documents, key=attrgetter('timestamp', 'title'))

#def sortTitle (sorts alphabetical order A-Z)
    #documents.sort(documents, key=attrgetter('title', 'timestamp'))

#def sortTag (sorts alphabetically by tag)

#def sortRelevant (option only available when filtering by tag. sorts by occurences of filtered tags)
```

```
#########################################
#Grid layout
#########################################

#def drawGrid(self, canvas):
    #for i in enumerate(documents):
        #row = i % cells in row
        #col = i // cells in row
        #cx = col * cellWidth + cellWidth//2 + margin
        #cy = row * cellHeight + cellHeight//2 + margin
        #draw document[i] at center of cell
```

Timeline Plan:

Day	Tasks
MON 8/3	Graphics layout, text editor functionality, rename document and change document timestamp
TUE 8/4	Tag documents and page content, filter by tag
WED 8/5	Sort documents by title (A-Z) and timestamp. Search document list or contents for matches.

THU 8/6	Streamline sorting and tagging functionality. Refine document editing and tag editing. Add copy paste from clipboard functionality into documents(?)
FRI 8/7	Add in extras, demo video

Extras:

- Tag color picker (rather than inputting a hex code string)
- Get machine time for timestamp
- Sort by relevance (when filtering tags, prioritizes documents with the greatest instance of that tag. If multiple selected, goes by alphabetical order of the tag name, then the count)

Version Control Plan:

I will be using **Github** to manage my project. At the end of every day, I will also back up my project to Google Drive.

****TP2 Update*****:

Design changes:
- Due to logistical constraints, I will not be adding word highlighting/tagging functionality. Instead, individual pages can be tagged
- The text editor may be fairly rudimentary. My stretch goal will be to implement insertion typing with cursor clicks, but this may not be in the final product.
  - Bare minimum, the text editor can be typed in (all letters and characters, plus newlines, tabs, spaces) and have backspace functionality.
- I have improved tag viewing UX. The thumbnails in the library will show the first three tags, but a button on the bottom menu will show all the tags on the document when clicked.

****TP3 Update*****:

No significant changes