Assignment 2: Evaluation Project and Report

Student Number: 215758
Module: G5026 | Human-Computer Interaction
Date: December 31, 2021

Introduction

Trilium[1] is a hierarchical note-taking program with backlinks that differs from traditional note-taking applications which are typically linear; it allows you to structure notes in a way that allows you to repeat less and navigate them easier. There are many features with more complexity versus more traditional software used for taking notes; it allows users to write scripts to add extra functionality from within the note editor itself, as well as add metadata, attributes to notes to help search them. We will be focusing on the main aspect of the program: note-taking. Some other note-taking software that is popular is Evernote[2], Microsoft Word[3], and Obsidian[4] which we will analyse first.

Background

Evernote

Feature	Positive compared to Trilium	Negative compared to Trilium
Add text to note	Acts like a typical word processor which is more universally understood	Each row cannot be easily customised to include a table or image vs trilium
Add images, files and to-do lists to notes	Doesn't store the image in a separate file which trilium does	Images are copied for each note rather than a reference to the image if reused vs trilium
Flexible organisation	Allows organisation that isn't strictly hierarchical	Is strictly hierarchical which helps with efficiency to get used to in trilium
Drawing / Sketch notes	Allows users to easily sketch freehand such as graphs	Allows a more programmatic way of creating graphs for accuracy in trilium
Search notes	Simple and easy to understand visually and isn't cluttered	Search is basic, only searches titles of notes while trilium search is

		advanced
Templates and template gallery	Simple and pre-defined to make it easy to create a new note with a template and has templates made by Evernote	Doesn't allow changing of structure, just attributes for notes which help to search and sort them in trilium
Text headers and customisation	Easily visible to change the colour of text, as well as add headers	Trilium has background text colours in trilium
Bullet, numbered and checklists	Easily visible and not hidden away, always in the top bar	More variety of list customisation in trilium

Microsoft Word

Feature	Positive compared to Trilium	Negative compared to Trilium
Single linear document	Easily understood and well used by most people who use a computer for work	The linearity is ridged which makes it harder to navigate
Text and text customisation	Change colour and size easily, sizes are integers rather than a preset	The UI for text customisation is not overwhelming with different options in trilium
Adding Math	Easily be able to insert math symbols	Uses TeX which allows math to be inserted easily compared to Word in trilium
Adding pictures and images	Easy, either copy and paste or upload	Can't re-use images without duplicating the data in another document
Bullet, numbered and checklists	Very customisable lists and easy to customise but no checklists	Checklists exist as a feature in trilium
Change font	Large font variety, easy to change in an accessible menu and can choose more than one font in a document	Allows more customizability through the user being able to add their own CSS file in trilium
Adding online videos and links	Easy to add videos from platforms like Youtube and Vimeo, and the same goes for links	Adding and editing links is easier than Word in Trilium through an edit link button next to the link
Text styles and headings	Many styles and headings options allow you to divide	A simplistic approach makes it less overwhelming in

	content within the document	trilium
Search	Searches for words within the one document makes it easy to understand and is simplistic	Search spans across multiple notes and allows more advanced searches using tags you can set in documents in trilium
Templating (Designer)	Allows preset fonts and colours for the document to use, not the structure of the document	Allows customisation of attributes for a note in trilium which helps to search

Obsidian

Feature	Positive compared to Trilium	Negative compared to Trilium
Backlinking	Text-based and a central feature of the program with a graph structure over the whole folder/vault	GUI based way of adding rather than markdown-esk. Higher up nodes in the structure automatically include child nodes in trilium
Adding text and text customisation	Adding text is much more like a word processor, uses markdown	Allows you to change text colour in trilium
Search	More simplistic approach but also allows you to search over many files with just words or attributes	More complex which allows a more specific search query for the data you want in trilium
Templating	Allows you to structure the format and tags, font customisation is exclusively in the theme	None, obsidian is better
Graphs	You can use excalidraw graphs locally, no need to learn a new syntax for creating graphs making it easier for the user to start making them	You have to install an addon in obsidian for this, and mermaid diagrams are more accurate than Obsidian
Inserting images	Has a central location with the images to be stored and reused from	You have to set up obsidian more structurally than trilium which allows a more lazy approach to their hierarchical structure
Structure	You can use it hierarchically but is more dependent on	The hierarchical structure is easier to set up for efficient

	backlinking which makes it more versatile	note-taking through navigation in trilium if the user doesn't decide to use a hierarchical structure in obsidian
Adding files	As the notes are stored in plain markdown in an easily usable structure the user can move files into the directory they want in their chosen file management software or drag and drop files, trilium uses a database	None, obsidian is better
Adding math	Uses TeX to insert math using inline editing	Inserting the equation through the GUI rather than typing it in as a word processor would do makes it much easier to add as a new user to the software
Bullet numbered and checklists	Uses markdown so the features are already implemented	More customisation and easier to use as you do not need to know markdown

Trilium

Trilium adds more complexity compared to the de facto software users use to write notes; with this added complexity there would be more difficult design choices to consider. Perhaps they overlooked an area due to focusing on one aspect of the program over another. Trilium follows a hierarchical structure and additionally backlinking to other notes which can break the somewhat rigid structure they use; due to this users may not understand how to use the software if they are completely new to the concept which could cause some usability issues too. The closest is obsidian which is well designed but also has some flaws, but doesn't follow a hierarchical structure. Trillium is also open source which could mean that no graphic designers who have experience in UX design were present when designing the interface which is another vector for why the UI could not work well for users.

Method

Trilium's user interface will be evaluated by three expert evaluators. The three evaluators include me and two others on the course, none of us have used this application before and will be completing tasks separate from each other.

We created a task list for the user to complete and a write-up specification for issues that violate a criterion in Jakob's Ten Usability Heuristics[5]. The task list is where certain actions should be performed which users typically use within note-taking software, as well as more specific to the unique functionalities of the software such as mermaid diagrams and the hierarchical structure of the notes.

The task list will ask the users to respond with a difficulty level from one to five, this will help with finding specific problems with the listed tasks through averaging the scores. If a task is more difficult than a one-level difficulty, we ask for them to justify why they had an issue with completing the task.

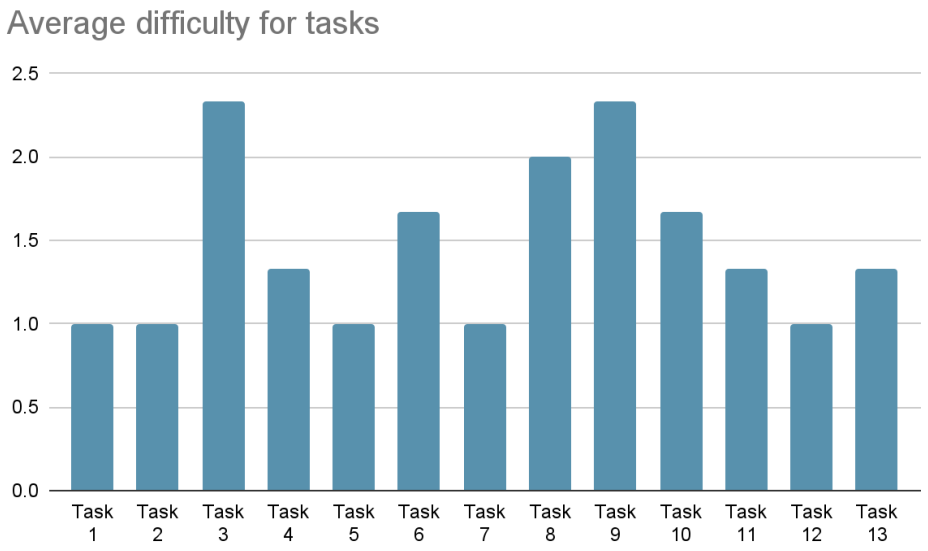
The task write-up will ask the expert to list any issues they have found which violates any of Jakob's Ten Usability Heuristics. This will help identify specific issues which users typically come across which affects the usability of the platform for them. They are given a link that has the heuristics listed which contains simple descriptors so they can pinpoint what one has been violated when they do find an issue.

The task list sent to evaluators is seen in the Appendix which contains the Tasks section.

Findings and Discussion

The evaluators' results can be seen in the Evaluator Results section in the Appendix.

Tabular and Graphed Data



#	Expert/s	Problem	Why	Heuristic violated	Severity	Fix suggestion
1	2	When trying to add a new note, there are two options, insert note after and insert child note	It may be confusing to novices using the app as they might not understand what a child is.	Help and Documentation	2	Add a help section or that introduces you to the language the application uses
2	2	It is not the best design to have the tutorial exist as a folder of notes	It requires the user to figure out the hierarchical system to efficiently use it	Help and Documentation	3	Have an interactive tutorial of the basics when first installed, offer advanced help if they want it

3	1, 2	The tutorial notes might get in the way for some users leading them to want to delete it	Once deleting the tutorial notes there's no way to get that help back	User Control and Freedom	2	Same as #2 fix suggestion
4	1, 2	There's no way to go back to a note you were recently editing	If someone accidentally clicked on a note they didn't mean to switch to, they can't go back to the previous note due to no way of going back	User Control and Freedom	4	Add a back button or history section to go back
5	3	There is not much feedback being shown on the interface, except essential information being shown.	A new user may not know if something was successful when an action was performed	Visibility of System Status	2	Add a small bar to the bottom of the application with a piece of confirmation text or a history bar of changes confirmed within the application
6	1, 3	Not all options are directly visible and some of them do take some thinking for them to be found	This may make the users' productivity lower and could be frustrating for them	Recognition rather than recall	3	Add a bar on the top of the page which is always visible
7	3	Not customisable unless you know CSS, changing fonts is in the settings menu but it's difficult to find and you have to check a box to enable overriding of fonts	Particular fonts or colours may make it hard to use the application such as for someone with dyslexia if the font doesn't help them	Aesthetic and Minimalist Design	2	Create a more accessible menu for changing fonts and don't hide it behind a confirmation box, as well as provide a selection of themes
8	3	No help documentation is present so if a user needs further help, they need to lookup information	Users may not know how to start using the program, notes are given as instructions but not so much on how to do it yourself, more like examples.	Help and Documentation	4	Add help documentation such as a mini offline version of their GitHub wiki
9	1	Folders are considered notes	Folders are not notes in the real world	Match between System and the Real World	3	Change the icon and language to a note/paper stack

#	Improvement	Why is this good?
1	Have an interactive tutorial of the basics when first installed, offer advanced help in a similar style but is optional; you can re-take them in the settings menu	This will introduce the user to the program giving a basic overview on how to use it, this helps one expert who raised the point that using notes to store how to use the notes isn't the best idea
2	Add a back button or not history section to go back	It allows users to navigate the application to a better degree through having navigation controls similar to a browser, this helps the second evaluator
3	Add a bar on the top of the page which is always visible for adding to notes	This helps two experts who had issues finding the not so visible elements you have to interact with for them to show up which you have to remember

4	Add help documentation which would be a mini offline version of their GitHub wiki for users to use within the application	This helps all experts who raised points about the lack of help or that the help given isn't helpful for complete novices as it shall give them a place to search their questions
5	Change the icon and language to a note/paper stack	This helps two experts who weren't too sure of the language of folders were when there was no direct way of creating them, and that folders were nested notes, more like a stack of paper
6	Add a small bar to the bottom of the application with a piece of confirmation text or a history tab for changes confirmed within the application	This allows the user to see that the action they have just performed has actually been done, this helps one user

In the write-up, we gained information from the violated heuristics and gained insight on what areas need improvement to comply with Jakobs Ten Heuristics. We have created a table with suggestions for improvements and why they are a good idea. The table is above this paragraph.

The task list results inform a better design for Trilium; we have found specific problem areas within the task list through the averaged tasks in a bar graph, particularly task three ("Create a nested note (folder)") and task nine ("Create a mermaid diagram"). The max difficulty peaks just below 2.5 which shows the application is successful but has some problem areas they need to work on. All evaluators pointed out the lack of help and documentation, evaluator one and two; evaluators one and two both had issues with navigating back to a previous note and added that it's not useful storing the instructions as notes within the platform as users will want to delete them. Evaluators three and one both had issues with the visibility of certain options not being forthright.

Conclusion

We discovered specific problem areas from the data gathered from the expert evaluators: that help and documentation is very lacking for this application and needs to be built upon an example being an interactive tutorial or an offline wiki for what element does what. Surprisingly users didn't struggle with the hierarchical structure however the language and visual language used was more so the main issue that confused the evaluators, an example being the folder icon being used for a nested note which doesn't match the real world due to the folder also being a note itself. Navigation was a slight issue too with backtracking to a previous note you recently finished. Tasks had an overall average of a difficulty of 1.46 which shows for the most part that the application is successful but there were tasks such as tasks three and nine which were just below a difficult score of 2.5 so it's still good but there are some things the developers and/or a UX designer could improve on.

Our limitations of the method we implemented are that we only used three users to test the UI, and these users are all computer science students who are familiar with a wide range of applications which gives them an advantage of already knowing common design patterns. We also only tested the note features only, not the scripting aspect or other more advanced features which could reveal more issues with the UI.

An extension for future research could be to include more features to test within the application such as the scripting feature, to add more people to test the application and

additionally use more inexperienced people to test the application rather than just computer science students.

Bibliography

[1] zadam, *Trilium Notes*. 2021. Accessed: Dec. 31, 2021. [Online]. Available: https://github.com/zadam/trilium

[2] 'Best Note Taking App - Organize Your Notes with Evernote', *Evernote*. https://evernote.com (accessed Dec. 31, 2021).

[3] 'Microsoft Word - Word Processing Software | Microsoft 365'. https://www.microsoft.com/en-us/microsoft-365/word (accessed Dec. 31, 2021).

[4] 'Obsidian: A knowledge base that works on local Markdown files.' https://obsidian.md/ (accessed Jan. 01, 2022).

[5] W. L. in R.-B. U. Experience, '10 Usability Heuristics for User Interface Design', *Nielsen Norman Group*. https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed Jan. 01, 2022).

Appendix

Tasks

Tasklist

1. Delete starter folder
2. Create a note
3. Create a nested note (folder)
4. Insert an image
5. Add a heading
6. Insert a math equation using TeX
7. Insert a code block
8. Encrypt the note
9. Create a mermaid diagram
10. Add a book note and some child notes with text in each
11. Include a note within a different note
12. Jump to a note
13. Upload a pdf file and view it

Task write-up

- Rate 1-5 for how difficult it was to perform a task, add a reason if above 1.
- List any issues found using Jakob's Ten Usability Heuristics.

Evaluator Results

Evaluator 1

Tasks:

1. 1
2. 1
3. 3: Initial language and icon shows a folder but to create a folder you have to nest two notes which seemed illogical at first, had to use the internet
4. 1
5. 1
6. 1
7. 1
8. 2: Encrypting a note was a toggle named protect which seems more like making something read-only rather than encrypting
9. 1
10. 2: Difficult to find at first
11. 2: Difficult to find
12. 1
13. 2: Should be able to just upload it, but I had to open up file viewer to drag and drop

Heuristics:

- **Visibility of System Status:** Only the necessary information is shown
- **Match between System and the Real World:** Works well for the most part however, folders are considered notes too which breaks how the real world works
- **User Control and Freedom:** Unable to navigate back to a previous note visited which can be limiting
- **Consistency and Standards:** Follows consistency and standards
- **Error prevention:** No errors were found, and if a potential breaking in the database may happen if you delete something it warns you
- **Recognition rather than recall:** Visibility isn't the greatest as menus to change text or insert particular elements such as a table are hidden until you remember the ¶ icon on the side to bring up the menu
- **Aesthetic and Minimalist Design:** Very minimalist but causes UI elements to be hidden which are important, and no easy customisation
- **Recognize, Diagnose, and Recover from Errors:** No errors came across
- **Help and Documentation:** No help was supplied except the initial folder which gets in the way so the user would delete it, you have to use their GitHub wiki or discussions section for help

Evaluator 2

Tasks:

1. 1
2. 1

3. 2: Wording of the drop down menu was confusing but soon figured it out
4. 2: Image had to be inserted as a file and wasn't clearly specified as image uploads as well
5. 1
6. 1
7. 1
8. 1
9. 2: Wasn't sure what a mermaid diagram was but it was easy to find
10. 1
11. 1
12. 1
13. 1

Heuristics:

When trying to add a new note, there are two options, insert note after and insert child note. These may be confusing to novices using the app as they might not understand what a child is in this context. This violates Help and Documentation as there is no embedded help for this task.

It is not the best design to have the tutorial exist as a folder of notes as it would require the user to figure out the hierarchical system to efficiently use it which makes it less accessible and violates Help and Documentation. It might also get in the way for some users leading them to want to delete it. Once deleting it however there's no way to get that help back which violates User Control and Freedom.

There's no way to go back to a note you were recently editing, e.g., If I accidentally clicked on a note I didn't mean to switch to, I can't go back to the previous note as there's no back button or note history I can access - this violates User Control and Freedom.

Evaluator 3

Tasks:

1. 1
2. 1
3. 2: Not as intuitive and direct as the previous tasks, had to think it through slightly more but still doable. Was not sure whether there was an option to make a nested note or whether I should've just right clicked in the general folder to add a nested note.
4. 1
5. 1
6. 3: This task was harder to complete compared to the other ones. Had to look up how to do it online and sort of had to go through the different options available. No direct option for it (no help option either). Was slightly confusing and misleading.
7. 1
8. 3: Not very simple to do this, but I wasn't sure how to do it at first.
9. 4: Was not sure where to find the option to do this and was quite confused. Not direct. Could not find the option to complete this task for quite some time.

10. 2: Simple to do, especially once the user has taken the grasp of how it works. Was not sure how to add a book note initially.
11. 1
12. 1

Issues:

There is no tutorial or many instructions and no help option with information that may be useful for the user to start using the application. Therefore, the user needs to get acquainted with the application and do tasks based on general knowledge acquired through other types of media or while using other similar applications. May not be suitable for all users. For example, someone who doesn't have much experience with apps like these, may have a hard time.

Heuristics:

- **Visibility of System Status:** There is not much feedback being shown on the interface, except essential information being shown.
- **Match between System and the Real World:** The application is simplistic and the language used is very easy to understand (except certain technical terms like “html” which would require some more prior knowledge)
- **User Control and Freedom:** Generally, it was easy to undo actions. The option to delete things is available which helps a lot with this aspect.
- **Consistency and Standards:** Follows general conventions and is quite consistent.
- **Error prevention:** Generally, the design prevents making errors but haven't really encountered any error messages while making use of the app.
- **Recognition rather than recall:** The interface is minimalistic which comes with its perks and cons. Not all options are directly visible and some of them do take some thinking for them to be found. I'd say it is quite average on this aspect.
- **Aesthetic and Minimalist Design:** Quite good on this aspect. The design is very minimalistic and has the main important option directly available. Makes use of conventionally known icons. Not really customisable though.
- **Recognise, diagnose and recover from errors:** No error messages occurred during the use of the app but generally the language used was simple, plain and direct – there were no error codes.
- **Help and Documentation:** No help documentation present so if a user needs further help, they need to look up information.