

GNG2101 Deliverable D

Smart Annotation Compilation

Submitted by

Section B Group 22

Bahar Ece Tuncyurek, 300051314

Carina Tam, 300105372

Adrian Kobelak, 300119469

Garvin Hui, 300114525

Ovais Azeem, 300112311

October 8, 2020

University of Ottawa

Abstract

Software designs were prepared for the client that were done based on concepts that were generated from the client's needs. After the client interview, notes were taken on how the designs could be improved. The result of the feedback is a user design prototype that combines the two designs that were presented to the client along with additional features that were suggested. In addition, a software prototype has begun, involving core backend functionalities. Tests were not conducted due to time constraints and the software prototype mainly functioning in the backend. Currently, the cost of prototyping and production of the final product are expected to be \$0 as the technologies being used to build the application are all open source. To prepare for the next client meeting, the goal is to have the first detailed design be fully constructed as a functioning prototype.

Table of Contents

List of Figures	iii
List of Tables	iv
List of Acronyms	v
1 Introduction	1
1.1 Background of the Report	1
1.2 Purpose	1
2 Detailed Design	2
2.1 Conceptual Designs	2
2.2 Client Interview	4
3 Critical Product Assumptions	6
4 Prototype 1	7
4.1 Software-Implemented Prototype	7
4.3 Prototype Testing	12
5 Plan for the next meeting	13
6 BOM	14
7 Conclusions and Recommendations for Future Work	16

List of Figures

Figure 2.1: Detailed design 1 part 1 - The note selection interface	2
Figure 2.2: Detailed design 1 part 2 - The note editing interface	2
Figure 2.3: Detailed design 2	3
Figure 4.1: Prototype design on the note selector	7
Figure 4.2: Prototype design on the file selector when adding media files	7
Figure 4.3: Mind Map Overview GUI	8
Figure 4.4: List View - Sort by Date GUI	9
Figure 4.5: List View - Sort by Importance GUI	9
Figure 4.6: Note popup view	10
Figure 4.7: Detailed Toolbar Layouts	10

List of Tables

Table 2.1: Feedback received on detailed design 1	4
Table 2.2: Feedback received on detailed design 2	5
Table 4.1: Expected and Actual Results from Software Prototype	12
Table 6.1: Bill of Materials (BOM)	14

List of Acronyms

Acronym	Definition
MS	Multiple Sclerosis
PDF	Portable document format
Subj.	Subjective
MB	Megabyte
UI	User Interface
UX	User Experience
GUI	Graphical User Interface
IDE	Integrated Development Environment

1 Introduction

1.1 Background of the Report

The basis of this project is to create a Smart Annotation Compilation, for Roksolana, a Master's student with Multiple Sclerosis (MS). This product allows for the user to keep track of their own notes and/or research findings, which will be of use as the client easily forgets information due to brain fog (as a result of MS). By using this product, the user will be able to connect concepts related to each other and present a thought train to effectively emphasize topics at hand. Previously, design concepts were generated, and a feasibility study was conducted. With the feasibility study, detailed designs were established for feedback.

1.2 Purpose

With the generation of design concepts completed, two detailed designs were created. A client interview was conducted to receive feedback on these designs and. The next steps are to create a first prototype and develop a bill of materials. This report will cover the detailed designs of the project, the prototype developed from the feedback of the detailed designs, and a bill of materials for the total expected cost of the project.

2 Detailed Design

2.1 Conceptual Designs

Previously, from the concept designs and the feasibility study, the solutions were refined into two detailed designs as illustrated below.

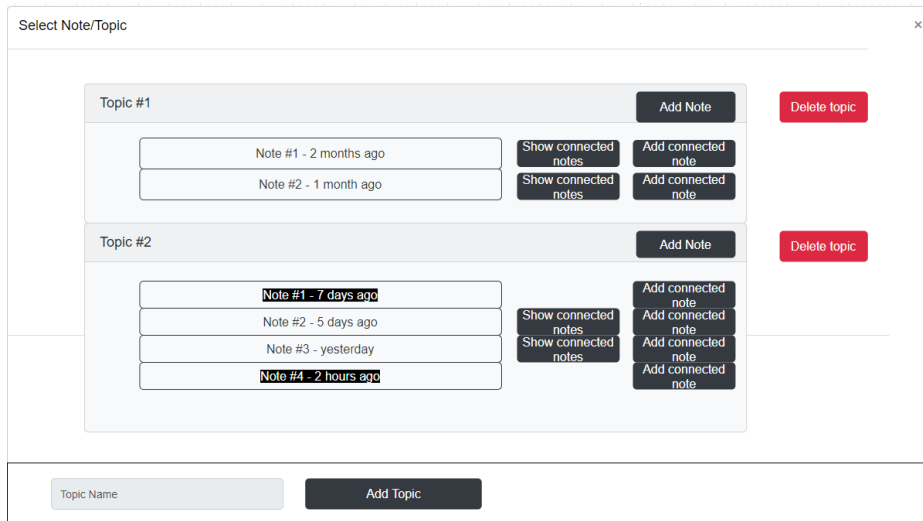


Figure 2.1: Detailed design 1 part 1 - The note selection interface

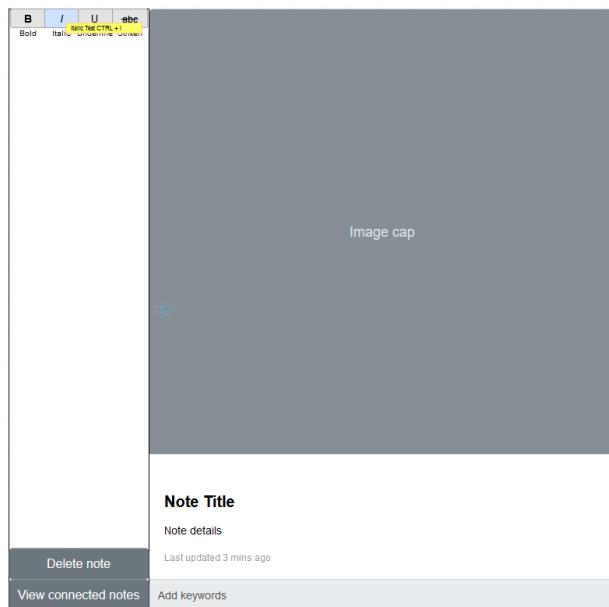


Figure 2.2: Detailed design 1 part 2 - The note editing interface

As shown above, the first concept has a simple interface where the note creation interface can be seen along with the date it was last updated. Next to each note, there's a button that enables the user to see which notes are connected/associated with it, to see if any other information is needed in order to facilitate

research or writing. The user is also allowed to insert a connected note, or to insert a brand-new note that is not connected to anything. Viewing the connected notes displays the corresponding notes with a highlight. With regards to figure 2.1.2, the idea was that when the user clicked on one of the notes from figure 2.1.1, it would open a window displaying this screen. The note itself would be composed of three main parts: the media, whether it be an image, or a PDF file, the note section, where the user would make annotations on the media displayed, and the keywords section, where the user can insert keywords based on what was written in the annotation, or based on the content of the media added.

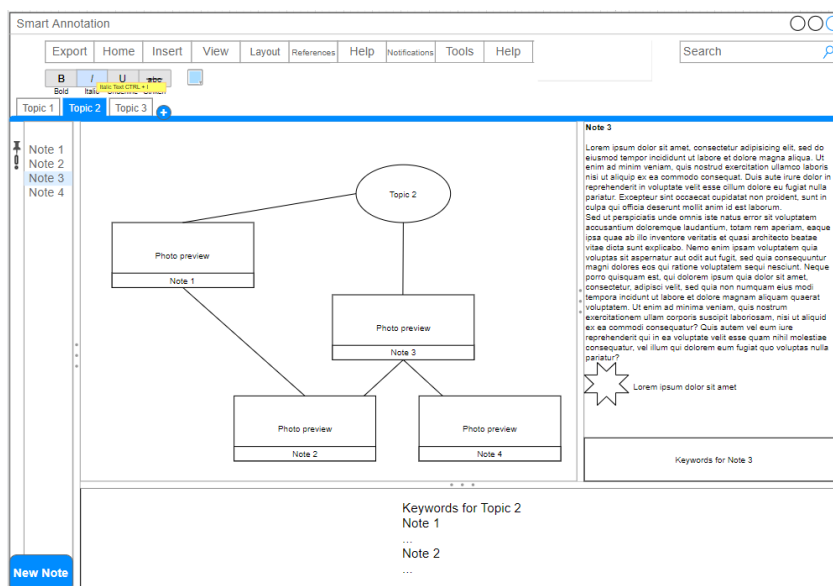


Figure 2.3: Detailed design 2

In terms of the second concept, the focus was more on the visual aspect of it, and what specific features the client wanted in it. The overall layout of the program is composed of a map/diagram, which visually shows how the notes link together, the notes sidebar, in order to facilitate navigation through the notes, the notes sidebar, where the user can see the note in full, the keywords section, and the toolbar. When coming up with this concept, it was thought of that the map could be scrollable, such that it can be an infinitely large canvas that the user can create. In addition, the user can also pin specific notes, and add markers into the note section. The user can also search for specific notes and/or keywords with the help of the search bar located at the top right of the window. The different sections would also be adjustable, such that the size of one part could be larger depending on the preference of the user. All of the additional

functionalities, such as setting notifications, exporting to PDF/printing, and inserting media is situated in the toolbar, similar to the arrangement present in Microsoft Office software.

2.2 Client Interview

During the client interview, both solutions were discussed for feedback. Overall, the client appeared to have a positive reception of the two designs that were proposed. However, there were still points of improvement to be made in both designs. The following tables outlines the feedback received on detailed design 1 and 2:

Table 2.1: Feedback received on detailed design 1

Concept	Likes	Dislikes	Points of Improvement
Note editor	<ul style="list-style-type: none">● Image takes a large portion of the screen● Layout is very well thought out● Adding keywords to personalize the notes● Seeing when notes were taken		
Note selector	<ul style="list-style-type: none">● Sorting by oldest to newest● Seeing when notes were taken	<ul style="list-style-type: none">● Highlighting to display notes that are connected● Using red to represent delete	<ul style="list-style-type: none">● Having more sorting filters would be better● Showing importance markers would be better

Table 2.2: Feedback received on detailed design 2

Concept	Likes	Dislikes	Points of Improvement
General UI	<ul style="list-style-type: none">• Menu bar• Search bar		
Crime board	<ul style="list-style-type: none">• Ability to rearrange and customize notes	<ul style="list-style-type: none">• Image preview probably will not be used	<ul style="list-style-type: none">• Different styles for links can help indicate different forms of connections• Timeline may be more useful than crime board• Maybe show note title and keywords instead of photo preview
Note Editor		<ul style="list-style-type: none">• Does not show image at large	<ul style="list-style-type: none">• Have the note editor pop up separately to increase working space
Note Selector	<ul style="list-style-type: none">• Importance markers next to notes	<ul style="list-style-type: none">• Cannot sort notes	

In general, both designs covered different aspects from each other that the client wanted to see in the final product. As such, the client expressed interest in combining the two ideas together. With this in mind, the prototype diagram in the following section has been adjusted accordingly. Additional feedback was also provided for general design as the prototypes continue to be developed. The client suggested to continue exploring ideas outside of what is considered normal to improve on the UI. Another suggestion was to improve the overall UI design in the sense of making it more visually appealing.

3 Critical Product Assumptions

The assumptions made through the development of the first prototype are as follows:

1. Assuming that the user must operate on a macOS laptop, and not on a tablet or a Windows machine
2. Assuming that the software must be a downloadable software, such that it is not a web application or an extension of another software
3. Assuming that all features must be accessible
 - a. Assuming that all colours used must be accessible
4. Assuming that the users' opinion of "looking attractive" is similar to that of the team
5. Assuming that the users will be satisfied with having only manual connections
6. Assuming that the client imagines the functionality the same as the team

4 Prototype 1

4.1 Software-Implemented Prototype

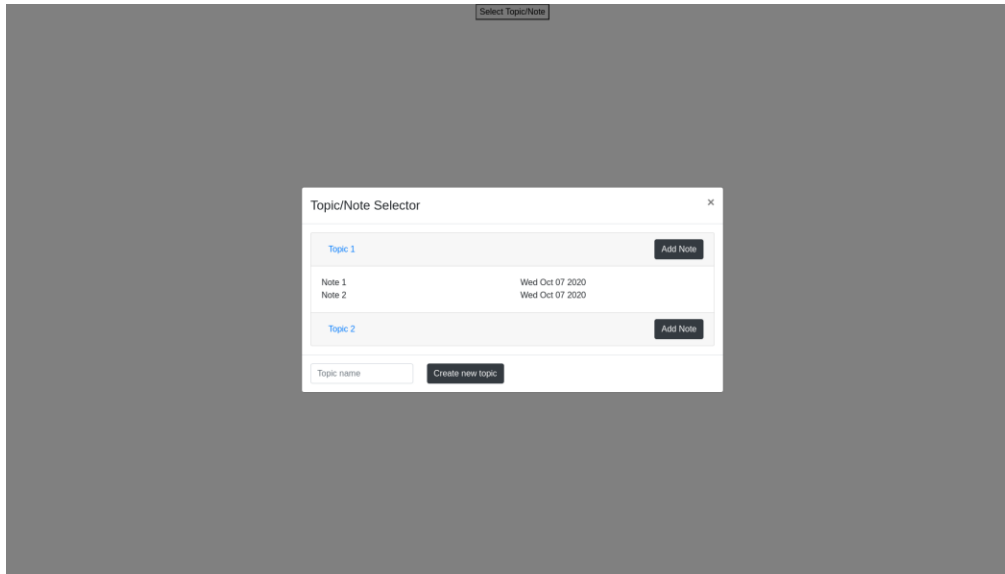


Figure 4.1: Prototype design on the note selector

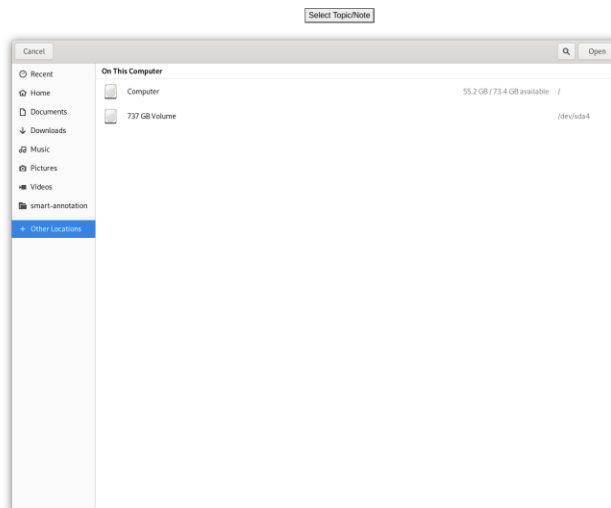


Figure 4.2: Prototype design on the file selector when adding media files

For the current prototype, there are three features that have been implemented, two of which are able to be visualized. The first feature is the note selector in figure 3.1 which is a direct translation of detailed design 1. Currently, the prototype version of the note selector is lacking certain features and will be updated as the project progresses. The second feature implemented in the prototype is uploading files into the application. Figure 3.2 displays the file explorer menu which is what users will interact with when

adding photo or audio files to their notes. The final feature implemented in the prototype is note saving. Most of the work done for the prototype thus far is in the backend of the application. This was done by identifying the common features that would be required in all possible solutions and isolating them into their own files. By doing so, the prototype could be developed in advance without having to worry about the feedback from the user, allowing for time to be saved.

4.2 User Interface Diagrams

In addition to commencing the implementation of the software, detailed user interface diagrams were also made to facilitate the implementation process, as well as to illustrate how the first prototype will most likely look like. The following figures will show the ideas generated from the team.

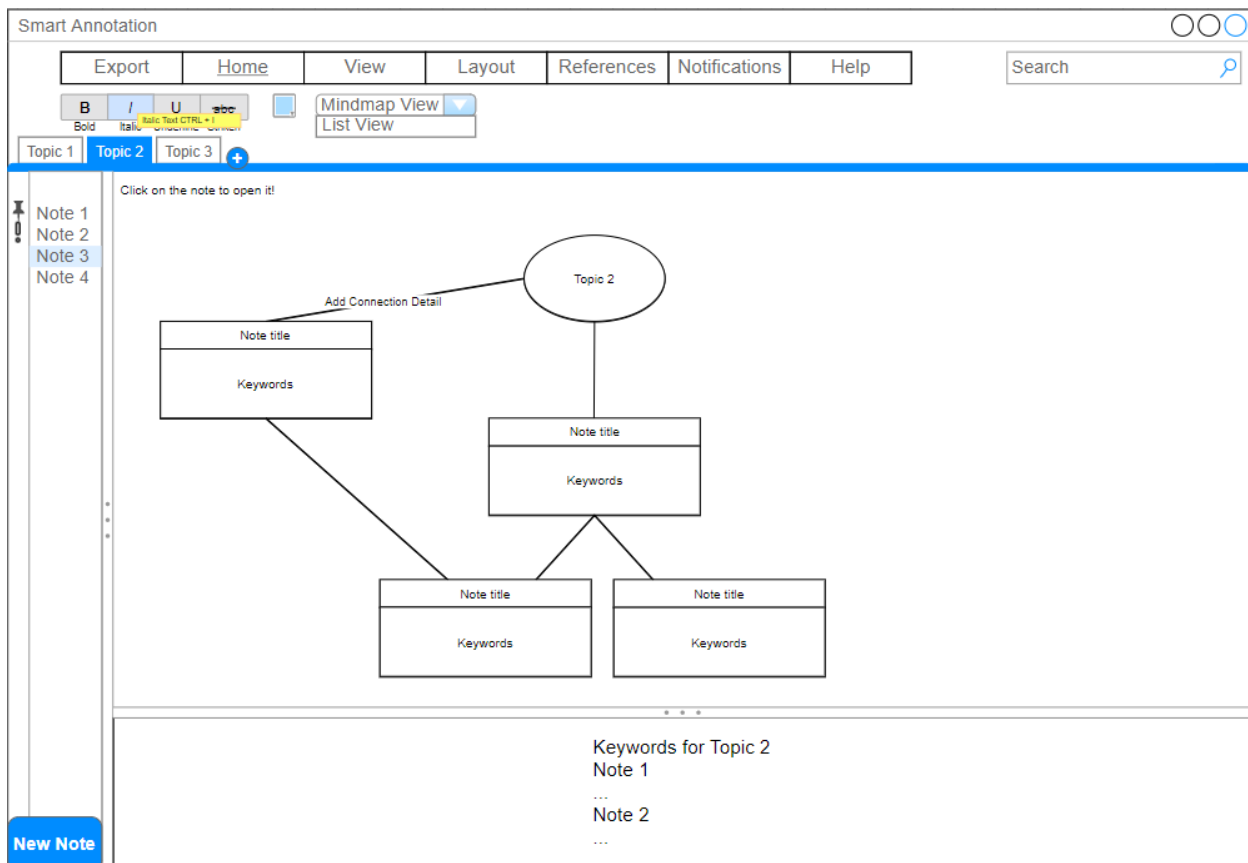


Figure 4.3: Mind Map Overview GUI

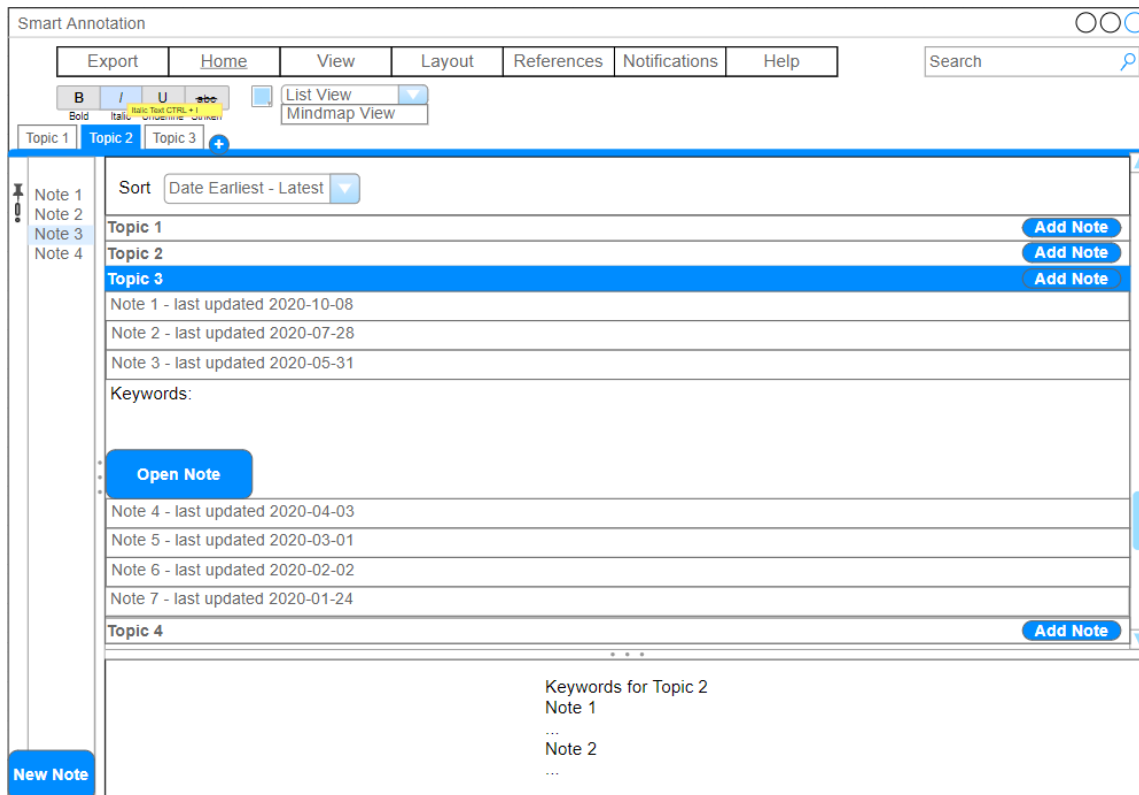


Figure 4.4: List View - Sort by Date GUI

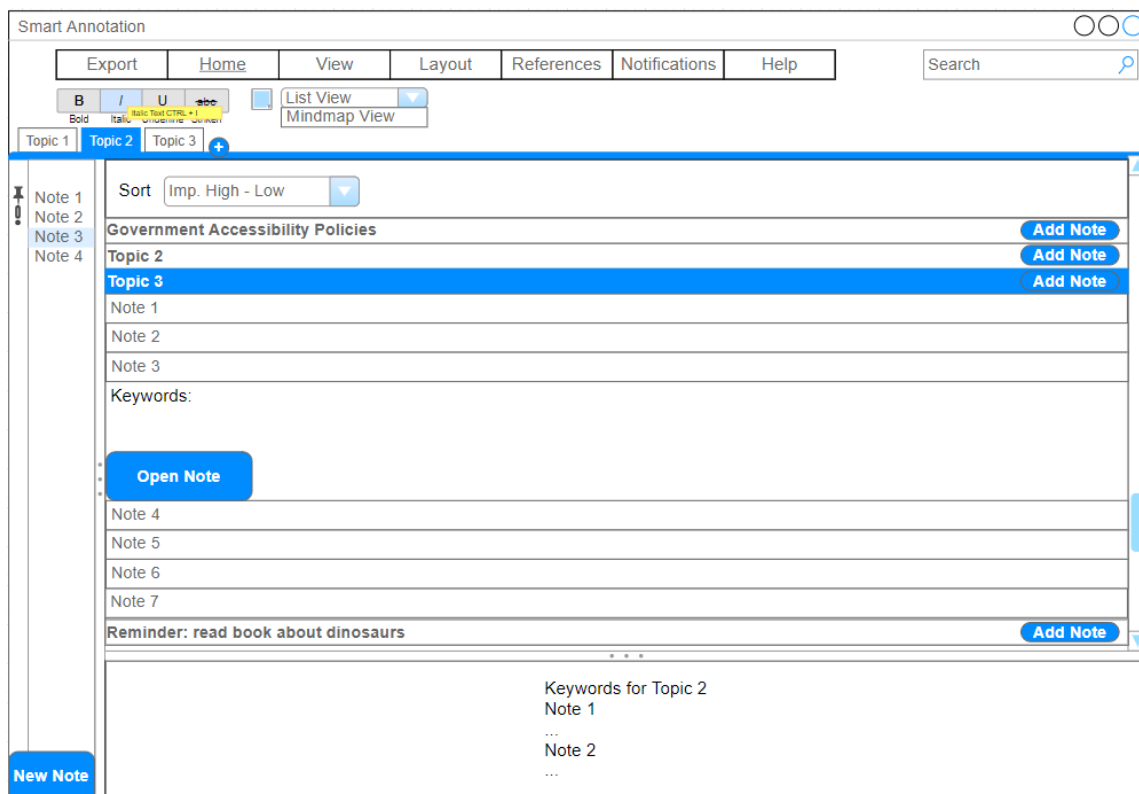


Figure 4.5: List View - Sort by Importance GUI

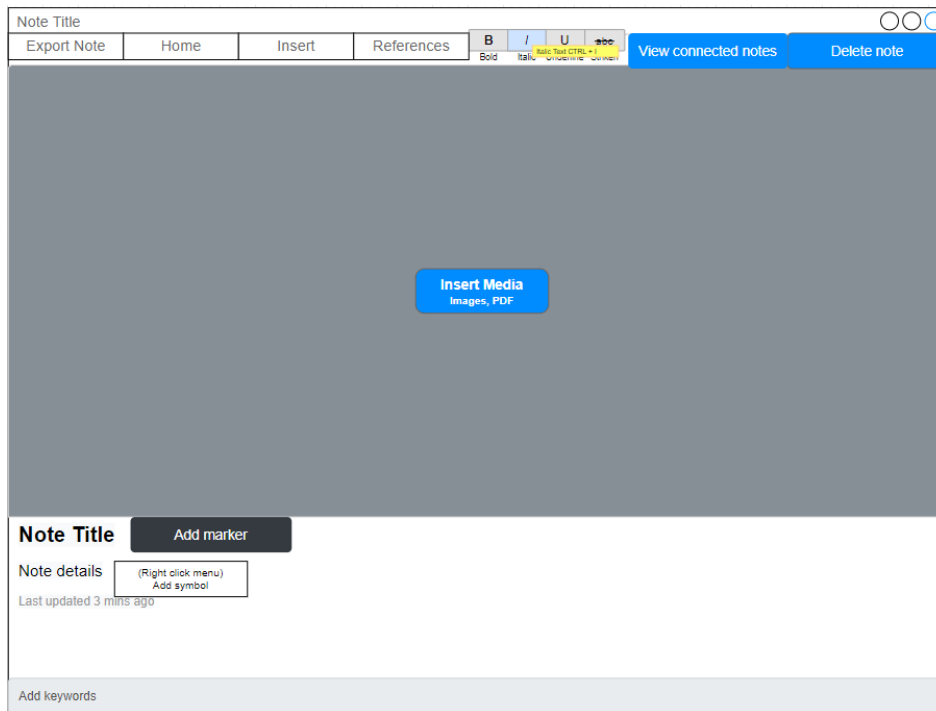


Figure 4.6: Note popup view

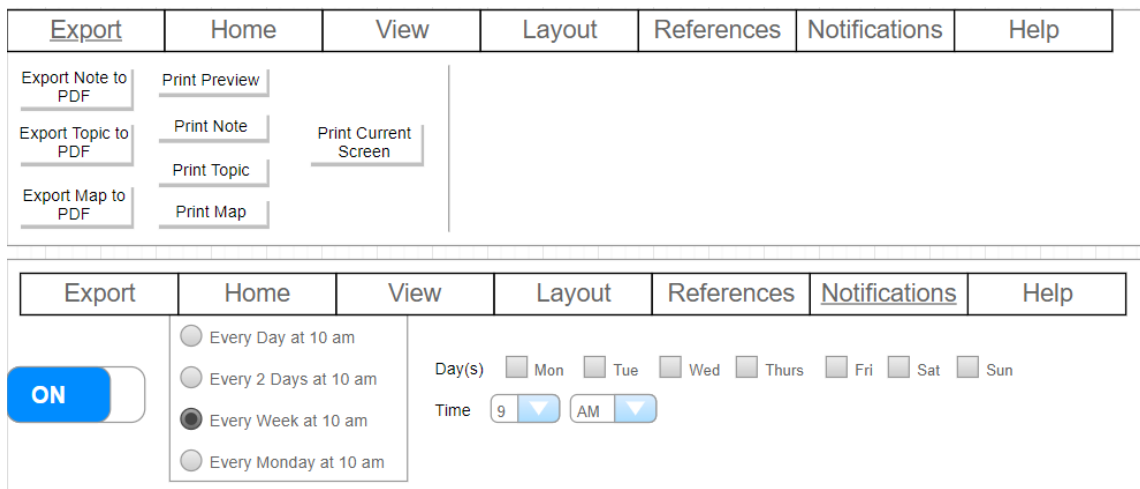


Figure 4.7: Detailed Toolbar Layouts

The user interface designs were generated based on the client feedback made on the conceptual designs. With reference to the client feedback, the following will be an explanation of what was added/removed from the previous designs. As there were features the client liked from both concepts, they were combined in order to produce a better product.

With regards to figure 3.2.1, in comparison to the conceptual design 2, the side note view was removed and replaced with a popup window that would display the note to facilitate viewing the note. The

popup window will have a similar interface to that from the first concept. Everything else remained generally unchanged due to the likeness of the client towards the appearance, however an option to change the view of the screen was added to add additional features the client suggested. While the main mind map page shows the overall connections map of the notes within a specific topic, the list view shows all the notes in a list format, similar to how regular notes resemble. Certain sorting elements were added as per the client's request, such as being able to see the notes based on the last date updated and seeing the notes in order of importance. Figures 3.2.2 and 3.2.3 outline how the interface of those would look like. As shown, the user will have the ability to see when the notes were last updated. In terms of sorting by importance, the user will have the option to label certain topics as important as they see fit and will be able to see them in order. The reasoning behind this sorting method is so that when the user is doing important research, they won't want to be seeing notes that are unrelated to what they are searching for or viewing unimportant information.

The note popup interface was largely inspired by the first concept, as the client expressed great happiness in how the layout was formatted. Some things were changed in order to make it more accessible, such as adding a button that can insert media instead of having it somewhere less accessible. Adding markers to the notes (to draw attention to certain points) was also made easier. A similar toolbar setup was also implemented in order to make the arrangement more familiar, as all note-taking and documenting tools have a toolbar nowadays.

In terms of the toolbar features, most of the tabs remain similar to those on Microsoft Word, or any Microsoft product. However, a few of them are different, and they will be explained below. The export tab allows the user to export notes, topics, and the main map to PDF, print them, as well as printing the entire screen present on the window. The notifications tab allows the user to set notifications to remind them to use the application. There are default options, such as reminding the user every day at 10AM, every two days, every week, and every Monday. However, the user can also set custom notifications, such as setting reminders on certain days at the same time. Some thought has gone into sending notifications, since like most programs, the platform used can only send notifications when the application is open. As such,

alternative options were thought upon, such as sending emails to the user reminding them, starting the application upon logging in to the computer, as well as having it set up such that when the user attempts to exit the program, it simply minimizes it, continuously reminding the user throughout their day on the computer.

4.3 Prototype Testing

Table 4.1: Expected and Actual Results from Software Prototype

Various Functions	Expected Results	Actual Results
Connecting Notes	Given a specific topic, the user can find related information to maximize their knowledge on the subject	The user can find information from related articles/notes about the research at hand creating a thought train
Sends Notifications	The # of times notifications remind the user can be customized to their preference	If the user has trouble remembering that the application exists, the notifications can be set to the duration of the user's memory
Markers	The user can insert markers to indicate importance and can help to find specific sections of notes/articles	Highlights info that the user intends to remember as important reminders in the future

Due to the limited time given to implement the team's first prototype (less than 2 days), testing was not possible. However, from now until the next client meeting, testing will be done in order to meet the client's standards and expectations.

5 Plan for the next meeting

During the next client meeting, there will have to be verification of the features' functionality mentioned earlier. There will be questions asked pertaining to the level of difficulty of finding specific topics and how long it took to find them. After analyzing difficulty of use, specific aesthetic improvements will be made. The client will be asked about the colour of the interface background confirming that there will be no problem for them to differentiate different buttons on the interface. The ability to customize will also be discussed and whether the client is able to change aspects of the interface to their liking. The client will also be asked how their issues with the layout can be solved if there are any to their preference.

In terms of having a physical prototype, the goal at minimum is to have completed what is shown in detailed design 1 with changes according to the feedback received. By doing so, the client will have a functioning piece of software they can begin to start using daily. In addition, the client will be able to test the features that were listed out in the design and can provide feedback in the event that the workflow or functionality of the application does not operate how the client expected.

6 BOM

Table 6.1: Bill of Materials (BOM)

Item Number	Part Name	Description	Quantity	Unit Cost	Extended Cost
1	NodeJS	JavaScript runtime	1	\$0.00	\$0.00
2	ReactJS	JavaScript frontend library	1	\$0.00	\$0.00
3	ElectronJS	Desktop application JavaScript framework	1	\$0.00	\$0.00
4	Visual Studio Code	Open source code editor	1	\$0.00	\$0.00
					\$0.00

As a result of this being a software project, the tools used were free. The tools can be downloaded from the internet from a price of \$0. However, after downloading the various software, they must be installed in order for the team to use it. The tools involved for the construction of the prototype include: NodeJS, ReactJS, ElectronJS, and Visual Studio Code. All this software can easily be downloaded from the internet and are free to use.

NodeJS is an open source server environment that can be run on various platforms (Windows, Linux, Unix, Mac OS X, etc.). It can create, open, read, write, delete, and close files on the server and uses JavaScript. Furthermore, the team is using node.js for the backend and uses asynchronous programming to run a unit of work separately from the main application thread.

ReactJS is used for making user interfaces or what the application looks like. It uses Node.js to compile and optimize the JavaScript code to create UI Test cases. Furthermore, the team is using React.js for the front end of the application and React.js uses JavaScript libraries. Also, it is open sourced and thus, the original source code is made freely available and can be redistributed and modified.

ElectronJS is an open-source software framework developed and maintained by GitHub. It allows for the development of desktop GUI applications using web technologies: it combines the Chromium rendering engine and the Node.js runtime.

Visual Studio Code is an IDE (Integrated Development Environment) which provides comprehensive facilities to computer programmers for software development. It is also a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, and had Git embedded in it. Furthermore, the team is using this IDE to program the standalone application for the client.

7 Conclusions and Recommendations for Future Work

After presenting various prototypes to the client, the client gave the team numerous points of feedback in order to improve upon the initial design concepts. Some of the feedback includes what the application should alternatively look like and what types of functionality to not include. Examples of functionality to not include would be having big fonts, highlighting certain parts of the document, and using certain colors like red and orange. An example of functionality to include in the next version of the application would include multiple forms of sorting.

Currently, the purpose of the first prototype is to emphasize the layout and its benefits with users. Later on, there will have to be further implementation in the functionality of the interface with more refined code. Given that the client was enthusiastic about both layouts that were created for the second client meeting, there is more freedom to decide on how the interface looks in the end.

By the next client meeting the first prototype will be ready and the client will be able to interact with it so that there can be more feedback on the ease of its use if there are any in order to generate the second prototype. Given that the client is glad that there are engineering students working on this task, she has communicated the fact that she is open to clarify her preferences for the interface and to let her know if there are any additional questions.