

GNG2101
Design Project User Manual

Annota: Smart Annotation Compilation User Manual

Submitted by

GNG 2101, Team B31

Philippe Gagné, 300056229

Jodi Qiao, 300100795

Daanish Khan 300126840

Rikki Romana, 300152011

Bjorn Viebrock, 300118186

December 10th, 2020

University of Ottawa

Abstract

People with disabilities or who are otherwise neurodivergent often have difficulty remaining focused and productive in their work, either due to physical disabilities including but not limited to reduced mobility and colourblindness, or due to psychological impairments like short attention spans stemming from ADD or ADHD. In 2020, as the world takes a sharp turn towards remote work, these problems are becoming increasingly exacerbated among our workforces.

With the help of our client, we found that there was a need for productivity software that would serve to alleviate some of these issues — notably memory and focus retention — while also appealing to users with wide-ranging accessibility needs.

The resulting product is Annota. Annota is a browser-based application that encompasses a suite of tools for personal project management and that is designed with accessibility in mind. Our priority was first to create tools that efficiently allowed users to track their documentation and overall project vision, but to make accessibility a core piece of the puzzle. Considerations were made for people with mobility issues by way of keyboard shortcuts and were made for people with visual impairments by employing a high-contrast, colourblind-friendly UI. Passive focus retention was quietly folded into the overall design by making Annota's custom home page the default new tab for the browser. In short, Annota is not just an accessibility platform. It is a productivity platform that considers accessibility in every design decision.

The existing build of Annota is a Flask-based web application that runs on Google Chrome from a locally hosted server. The primary technologies it employs are HTML, CSS via the Bootstrap framework, JavaScript (including the graph theory library Cytoscape.js), and Python via Flask.

Annota is a project with much room to grow. For future students looking to build on this version of Annota, we would recommend that they consider implementing new features that tie in to the existing functionality, integrating a full back-end and database, as well as increasing the accessibility of Annota.

Table of Contents

<i>Abstract</i>	<i>i</i>
<i>Table of Contents</i>	<i>ii</i>
<i>List of Figures</i>	<i>iii</i>
<i>List of Tables</i>	<i>iv</i>
<i>List of Acronyms</i>	<i>v</i>
<i>1 Introduction</i>	<i>6</i>
1.1 Problem.....	6
1.2 Fundamental Customer Needs.....	6
1.3 Solution.....	9
1.4 Key Aspects of Product	10
<i>2 How the Prototype is Made</i>	<i>11</i>
2.1 BOM (Bill of Materials)	11
2.2 Tools/Equipment List	11
2.3 Instructions	12
<i>3 Functions and Capabilities</i>	<i>13</i>
<i>4 How to Operate</i>	<i>18</i>
<i>5 Business Model</i>	<i>18</i>
<i>6 Conclusions and Recommendations for Future Work</i>	<i>20</i>
<i>7 Bibliography</i>	<i>21</i>
<i>APPENDIX</i>	<i>22</i>

List of Figures

<i>Figure 1: Overview of Annota</i>	10
<i>Figure 2: Main Webapp Navigation/Functionality Flowchart.....</i>	13
<i>Figure 3: Chrome Extension Navigation/Functionality Flowchart.....</i>	14
<i>Figure 4: Map View Functionality Flowchart</i>	14
<i>Figure 5: Timeline View Functionality Flowchart</i>	15
<i>Figure 6: Annotate Functionality</i>	16
<i>Figure 7: Extension Context Menu and Commands</i>	16
<i>Figure 8: Annota home page</i>	17
<i>Figure 9: View pages with document selection</i>	17

List of Tables

Table 1: Bill of Materials..... 11

Table 2: List of software frameworks and libraries..... 11

List of Acronyms

Acronym	Definition
API	Application Programming Interface,
UI	User Interface Design
UX	User Experience
OCR	Optical Character Recognition
BOM	Bill of Material
JS	JavaScript

1 Introduction

1.1 Problem

A discourse analysis, the study of written or spoken language in relation to its social context, requires taking notes and putting annotations in documents and images to be able to analyze the data. The client for this project, is a second year Masters student with a disability and chronic illness, is seeking to perform a discourse analysis of a government bill for her research, but she is not always capable of remembering the documents and what ties the have to other pieces of information. The purpose of this project is to create a way for her to keep a visual timeline of all the annotated and reviewed documents and how the information might flow from one to another.

1.2 Fundamental Customer Needs

The customer needs played an important role in the conception of the prototypes. In order to assess these needs it was important to emphasize with the customer. This was done by asking open ended questions, asking a lot of questions and examining the body language. From the answers of the client the following customer needs where determined:

Table 1: Customer Needs Statements, Interpretation, and Importance Ranking

Category	Customer Statements	Interpretation	Importance
Functionality	“I want an application that links documents together.”	Information from many different documents and software systems must be visible from a single space	5
	“I would like for it to work like a conspiracy corkboard with sticky notes from a crime movie.”		5
	“I want the application to allow me to annotate documents.”	An included browser extension will allow text from documents to be highlighted and annotated.	5
	“I’d prefer for it to be compatible with Apple Pages, but	To maximize the audience and reduce development time,	4

	Google Docs is also okay”	we're aiming primarily for Google Docs compatibility. Pages is also in consideration but will take the backseat.	
	“Zotero integration would be nice.”	The Zotero API will be used to relay information between our app the user’s instance of Zotero.	3
	“I’d like to have a visual timeline of the documents and annotations I’ve recorded”	There will be a visual interface in the web app to display all the documents and annotations. It will include an overview of all the primary documents and a focused “timeline” view for each document.	5
	“I need the application to be compatible with Apple devices, but I’d like it to work with Windows as well.”	This should be a non-issue, as it will be designed as a Google Chrome browser extension.	1
	“I want the application to have an easy-to-use UI.”	UI/UX decisions will be based on intuitive visual clues that clearly communicate the relationship between pieces of information. Minimalist designs will be employed to reduce visual clutter.	5
	“I need the software to work with multiple types of documents (PDF, Website,	There will be a feature to include custom documents that are difficult to	5

	Books, E-Books, Excel sheets, images).”	access virtually, such as books, e-books, and images. An OCR (Optical Character Recognition) system will be tied in to convert pictures of text into PDFs that can be highlighted and annotated.	
	“I’d like to have an annotation classification system”	Annotations will have various priority classes, with customization (e.g. color, name, symbols)	5
	“I want the application to remind me to do work that day, and for notifications to be customizable.”	A quiet push notification that appears in the browser at a user-provided time could be implemented.	3
Accessibility	“I do not want a Duolingo level of notifications”	Allow users to control the notifications.	3
	“Avoid colours for communication, use more symbols.”	The system should work just as well with colours as without.	5
	“I’d like for there to be a no-mouse option.”	Keyboard shortcuts can be implemented to trigger key features.	2
	“I would like to read the words clearly/have large fonts.”	Font size and high-contrast settings will be available.	5
	“I want text-to-speech”	A text-to-speech button will be implemented so that annotations and notes can be heard aloud.	4

	“I would like for there to be colour blindness options.”	Settings to accommodate users with Protanopia, Deuteranopia, and Tritanopia will be included. If time permits, custom color schemes are also a possibility.	5
--	--	---	---

Through analyzing the client’s needs and our design criteria for the software, our team developed this Problem Statement:

A need exists for a second-year Masters student with disabilities and chronic illnesses, to have access to accessible software solutions that would alleviate the effects of her disability as they relate to her ability to do her work. These effects include colour blindness, forgetfulness, and easy distraction. The client is looking for tools to organize, annotate, and display her work and reference documentation [1].

1.3 Solution

To solve this problem, we created Annota, an accessibility-forward virtual productivity environment for Google Chrome. Annota is a webapp and Google Chrome extension that includes a suite of online productivity tools to give people new ways to manage school and work projects. Annota ties together unique file visualization tools, called Maps, and dynamic annotation tools into a unique virtual productivity environment, bolstered by an intuitive, user-friendly interface. The anchoring goal behind all these features is to maximize people’s productivity throughout the day by avoiding distractions and staying in their workflow.

For Annota, we had come up with target specifications that we felt were important to the success of the application. The app must be easy to use and be operable by individuals with disabilities, such as colorblindness. The application must have notifications that can be customized by the user, as well as including complete interoperability across all operating systems. Annota should be able to integrate with external software (such as Zotero) and should be able to parse any type of document into the webapp.

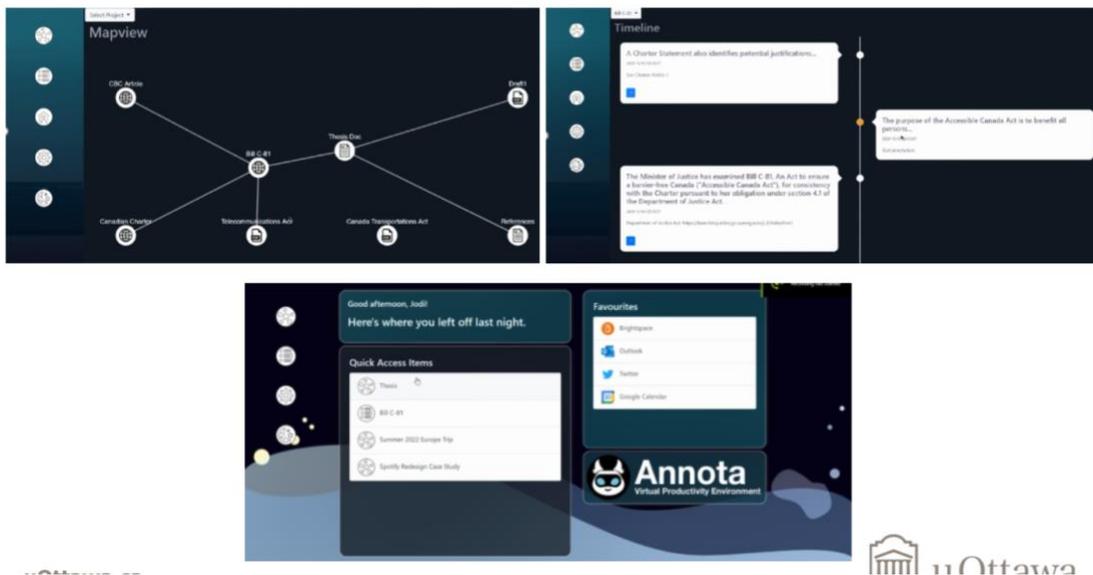


Figure 1: Overview of Annota

1.4 Key Aspects of Product

While there are many individual online productivity tools in the Google Chrome web store right now, there is no product in the market like Annota as it combines several productivity tools into one Google Chrome extension. Additionally, the product stands out from others because of the accessibility features that has been incorporated into Annota. It was important to our team that as we move into an increasingly digital workspace nobody gets left behind, especially not the people who are already in the margins of our work culture. With this mission in mind, we've taken steps towards alleviating issues surrounding colour blindness, movement limitations, as well as forgetfulness and short attention span [2].

2 How the Prototype is Made

2.1 BOM (Bill of Materials)

Since this is a software project, we used mostly free, open-source resources to build Annota. Some software frameworks and libraries are free to use, even in commercial applications, if we keep our source code open source. The following table shows the bill of materials for this project [3].

Table 1: Bill of Materials

Component	Description	\$(CAD)
Chrome Web Store Developer Registration	One-time upfront payment required to register as a Chrome developer, in order to publish a Chrome extension	5 USD
Adobe Illustrator	Graphics and vector design software, which may be used to make website graphics.	33/mo Student tier subscription (This was free for us as Jodi already has it due to her design side gig)
Total:		~5

2.2 Tools/Equipment List

In addition to the components mentions in the BOM, the following is a table that shows the tools and libraries that we used to create Annota.

Table 2: List of software frameworks and libraries

Adobe XD	UI and UX design and collaboration tool from Adobe. Used for drawing visual concepts, wireframing and constructing Prototype 1.
Flask	Flask is a free micro web framework that we can use to build our web app's backend.

Document Object Model (DOM)	The DOM framework for frontend development will be made using HTML, CSS and JavaScript (JS).
Bootstrap	Bootstrap is a frontend framework with reusable HTML, CSS and JS components.
Bootstrap Studio	Bootstrap Studio is a WYSIWYG web design tool for the Bootstrap framework. Will be used to generate HTML, CSS and JS code.
JQuery	jQuery free JS library that we may use as glue code.
D3.js	D3 is a JS library that manipulates documents based on data. We will use it to build our timeline and web views.
Plotly.js	Plotly is a JS library that is used to build data visualizations. We will use it to build our timeline and web views.
Vis.js	Vis.js is a dynamic browser visualization library. We will use it to build our timeline and web views.
React JS	ReactJS free and open source frontend JS library for building UI.
SQLite	SQLite is a small, fast, open-source and self-contained (no cloud/server connections, simply writes to user's disk) SQL database engine. It can be used together with Flask through flask-sqlalchemy.
Docker	Docker is a container we can use to package our software so the client and any other potential users can install and run our app on their devices.

2.3 Instructions

In the following GitHub repository, it shows the codes and instructions on how we built and designed the software.

<https://github.com/B31AccessibleAnnotations/B31-Annota>

3 Functions and Capabilities

The following figures shows the functionality flowcharts of the main features of Annota.

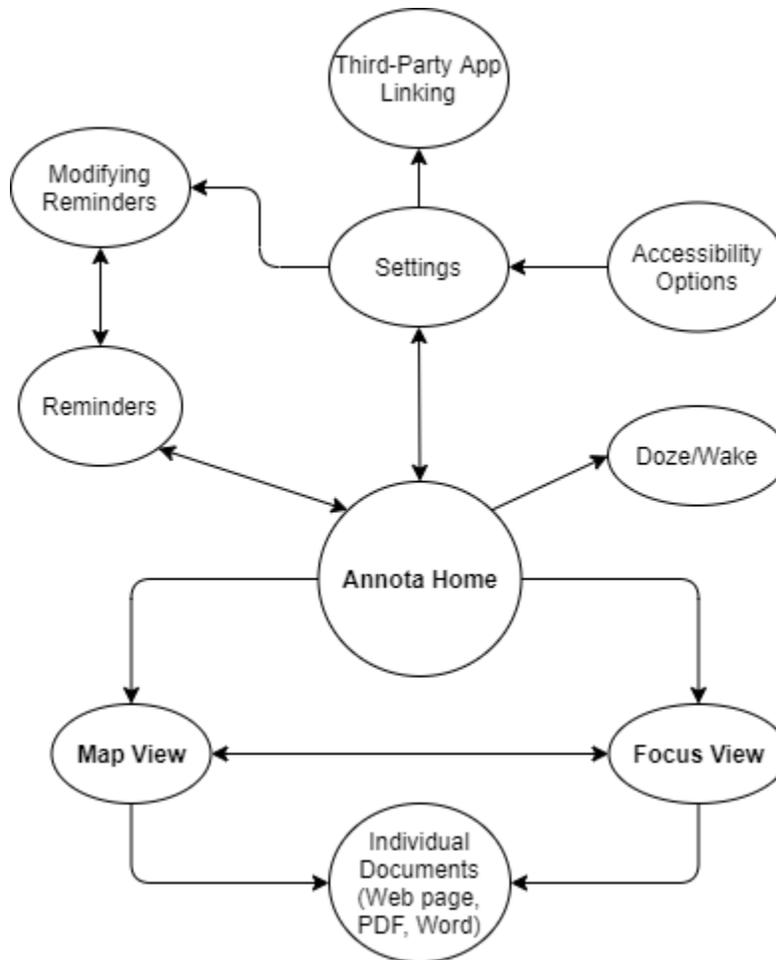


Figure 2: Main Webapp Navigation/Functionality Flowchart

Annota launches the Annota Home page on browser start-up. From here, users can view their Reminders and access a variety of Annota features. User preferences can be tailored to make Annota Home the default page whenever a new tab is opened, thus keeping the user tied into the Annota productivity space continuously.

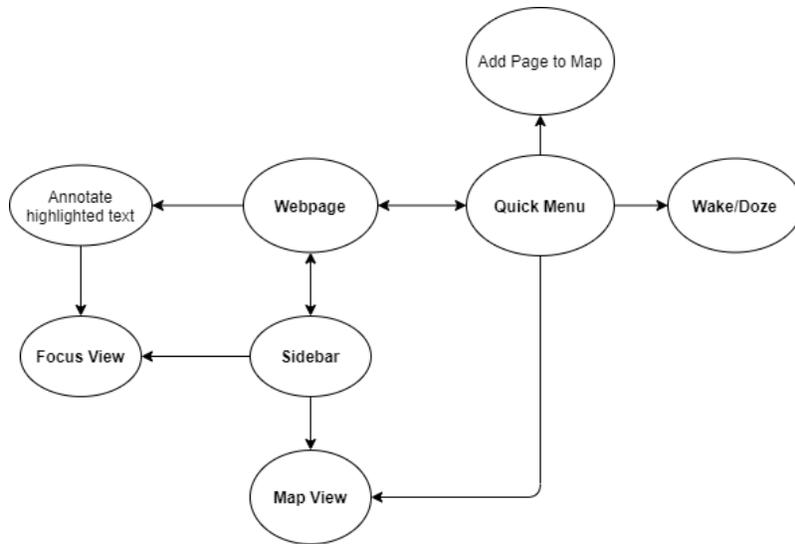


Figure 3: Chrome Extension Navigation/Functionality Flowchart

When in webpages, users can highlight text to annotate, which automatically adds the page to the Map, or they can add the page to the Map via the Quick Menu without making any annotations. The Annota Sidebar is always active, and can be pulled out from the left side of the browser at any time to provide quick access to key features (notably Map View, Annota Home, Settings).

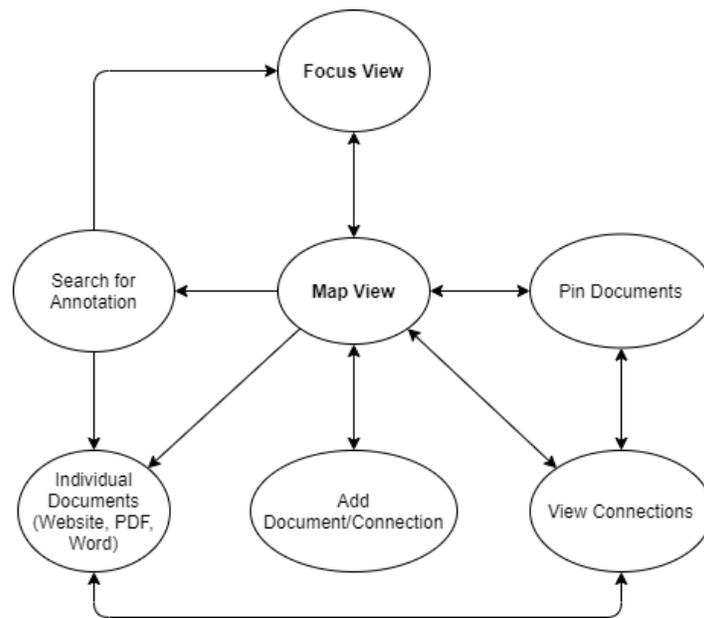


Figure 4: Map View Functionality Flowchart

Map View is where users can visualize all their documents in one place. When a document is added to Annota, it can be seen in a separate list menu, and from there it can be pinned or connected to the Map. Clicking on a document in the Map will open that document's Focus View.

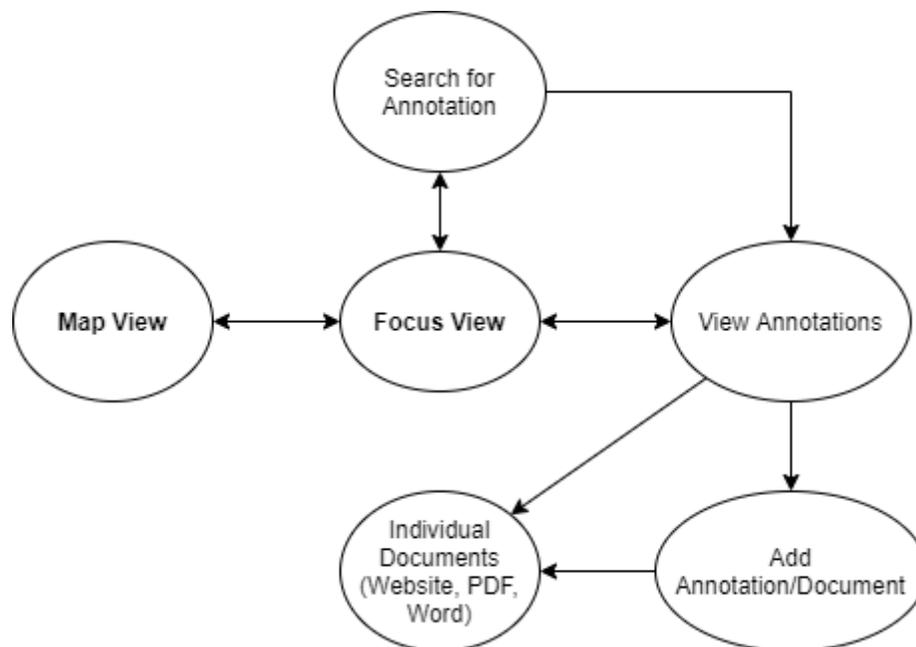


Figure 5: Timeline View Functionality Flowchart

In Focus View, users can view and edit annotations for a given document. Annotations are displayed in a “timeline” spanning the length of the document. They are marked with symbols and colours, and can have attached hyperlinks or files.

The following figures below show the screenshots of the accessibility features of Annota [4].

3. Document your latest prototype(s) using as many sketches/diagrams/pictures as required and explain the purpose and function of your prototype(s).
4. Carry out prototype testing, analyze and evaluate performance compared to the target specifications and prototype specifications. The document shows expected specifications to the one your design.

Now that a first prototype is ready, you can start testing prototypes and/or begin to document your design and other requirements.

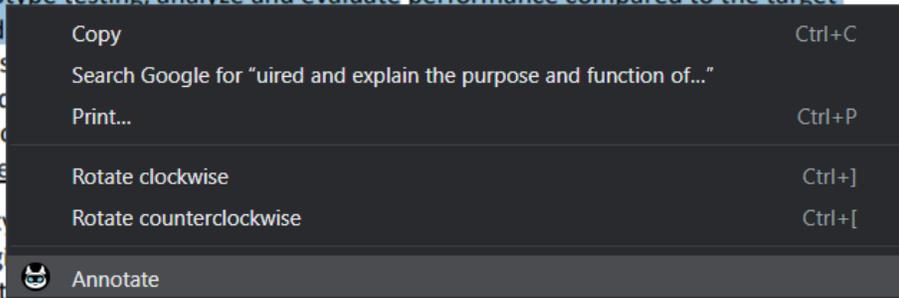


Figure 6: Annotate Functionality

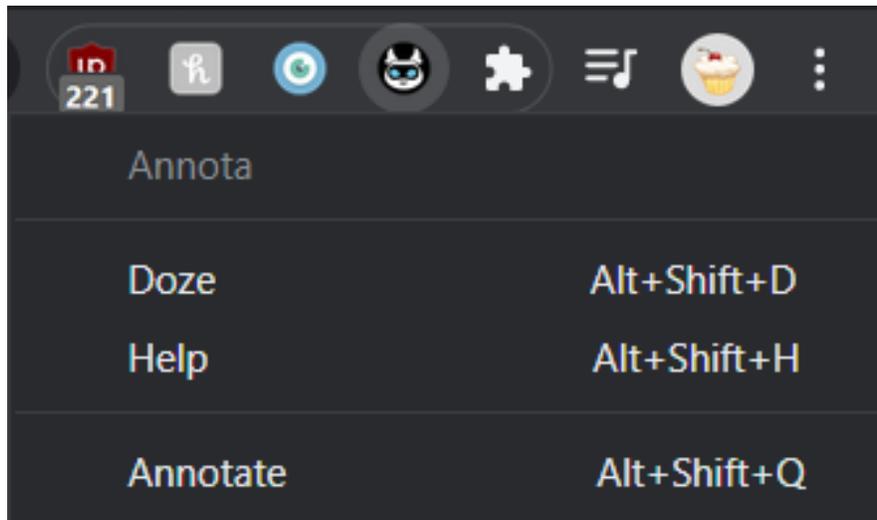


Figure 7: Extension Context Menu and Commands

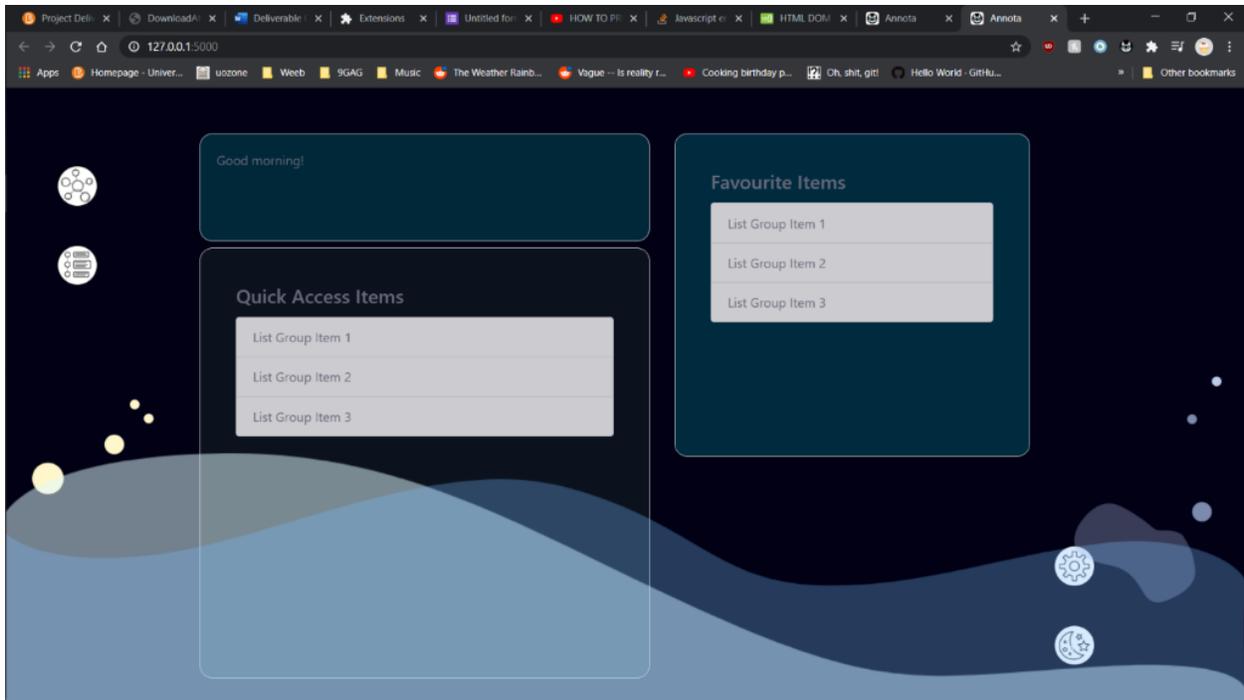


Figure 8: Annota home page

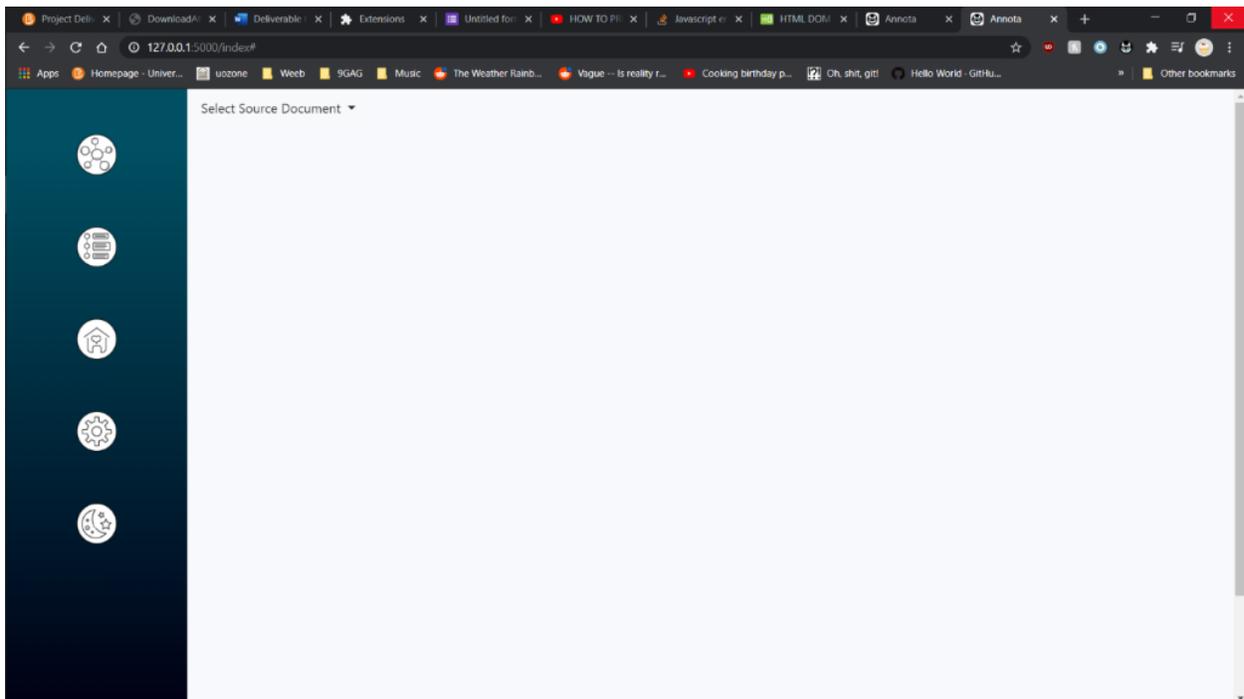


Figure 9: View pages with document selection

Additionally, the following video shows the demonstration of Annota:

https://www.youtube.com/watch?v=oJfJi673Pes&feature=youtu.be&ab_channel=RikkiRomana

4 How to Operate

Critical requirement: current version of Python3 installed on your computer.

1. Install the zip file provided in the GitHub repository:

<https://github.com/B31AccessibleAnnotations/B31-Annota>

2. Extract the file to desktop.

3. Open google chrome, and go to chrome://extensions -> Load unpacked

4. Navigate to the directory you have Annota in, and click annota-extension

5. Load that extension.

6. Download and install this extension (Colorblinding):

<https://chrome.google.com/webstore/detail/colorblinding/dgbgleaofjainknadoffbjkclibbgaa>

(for optional extra colourblind testing)

7. Open Windows Powershell

8. Copy Annota's directory

9. In powershell, type cd <paste_directory_here_dont_include_the_brackets>

10. Type ./backend-init.ps1 and hit enter. This will start up the backend for the extension.

11. Open a new Chrome window.

12. Hit Alt+Shift+H to open the help page.

5 Business Model

To commercialize our product, we have chosen to follow a “**freemium**” subscription-based business model. Because Annota is hosted on the Google Chrome Store, which provides ecommerce infrastructure and a variety of many monetization options, it should be relatively easy to implement this kind of subscription service. Annota is a multi-faceted product with ample room to grow and with a variety of quantitative features (quantity of annotations, number of documents, storage space, etc), which means that we also have lots of flexibility in which parts of the product we choose to leverage for monetization.

The intent is for that when the user first downloads our product, they will be able to access a simplified version of Annota with limited features (that remain undetermined as of the writing of this manual). This version of Annota would be totally free and would serve, in essence, as the

“hook” for Annota. The unrestrained version of Annota (known tentatively as Annota Premium), with all functions and features, would be locked behind a subscription paywall. Our early drafts place this subscription at \$1.49/mo, with a year-long subscription available at a reduced price of \$15/year.

Part of the marketing strategy for Annota Premium would also be to provide free 1-month Annota Premium trials during which users would have full access to Annota’s suite of features. When the trial is over, users would have the option of either continuing to use Annota Premium with monthly fees or to return to Annota Basic.

It is important that Annota, especially in its early stages, be available free-of-charge on some level. Part of Annota’s growth strategy is to build a userbase through word-of-mouth, which is very difficult to make happen if there is an immediate paywall upon downloading the application. With a free edition, we can entice users into making Annota part of their daily productivity methods before they feel like they need to upgrade past Annota Basic’s limitations (ex: storage space). In a sense, we want to reel in the customer by making their workflow dependent on Annota. Having two price points for the subscription is also an important part of the business model. Having a yearly subscription at a lower price is a win-win for the Annota team and for consumers. We expect more users to opt for this yearly payment as it is the cheapest option for them, while also creating more revenue upfront, which means means that we can reinvest into Annota’s development quicker and encourage rapid growth.

Distribution Approach

Our intent is for Annota to be distributed as an extension on the Google Chrome Web Store. Google Chrome Web Store already provides ecommerce infrastructure for the extensions hosted on it, which makes this relatively easy to implement without much overhead cost. Through the Chrome Store we can modify the free trial length as well as the price of the subscription. The other side of this coin, however, is that Google takes a cut of each conducted transaction.

As of the writing of this manual in December 2020, the Annota prototype is not yet available on the Google Chrome Web Store. If anyone working on Annota in the future is considering officially distributing Annota on the Google Chrome Web Store, we would recommend them to only do this if they can also implement cloud data storage for the application. Although it does interface with Google Chrome as a means of display, Annota currently relies entirely on the client’s infrastructure (i.e. the users’ own computer memory) for data storage upon installation.

Business Model Interaction

Our business model is one-to-many, where we are the sole service provider to our customers without need for middleware or becoming our own platform.

Monetization Ethics

Although Annota currently does not have any monetization built into it, we would ask future contributors to follow certain design and ethics standards in the monetization of the product. Firstly, we would request that Annota remain ad-free. We’ve already set the groundwork for other forms of monetization that we feel are very fair, and these decisions were based partially on a collective agreement in the foundational team that ads would be detrimental to the user experience of Annota. Although there is no ethical basis for this decision, we do feel strongly

about it as it relates to the usability of the product and would request that future contributors respect this feeling going forward. Secondly, we believe strongly that accessibility as it incorporates into Annota should be free of financial barriers. We disagree, on principle, with the commodification of accessibility as it financially exploits marginalized members of our society for factors that are outside of their control. It is for this reason that we would request that **no accessibility features in Annota be paywalled**. Annota is a product with much room for expansion, and there a multitude of perfectly valid ways of monetizing it without compromising this principle. Thank you for respecting the foundational team's wishes on this matter.

The following video link shows our 1-minute business pitch for Annota:

https://www.youtube.com/watch?v=-gtoeTOzxD8&feature=youtu.be&ab_channel=TastyPastry

6 Conclusions and Recommendations for Future Work

One very important lesson that we learned during this project is that accessibility should not be a gimmick. We set out to create a product with the intent of appealing to the need for greater accessibility online, and we could have easily created something that met only that purpose, but what we found is that it was a much more fulfilling experience for ourselves and for users to work with a tool that had a more meaningful purpose but that also incorporated accessibility as a foundational principle of its design.

The web has enabled people with disabilities to have more independence in their day-to-day lives, but with that we need to keep in mind that the design of a website or app can make or break how usable it is for them. Currently, many essential online tools are lacking in accessibility options, but it really doesn't have to be that way. As designers and developers, it's easy to rely on third parties to find the solution to these problems, but if we continue to work this way, we'll soon find that the internet experience for many of us will have become a cobbled mess of third-party extensions that attempt to put band-aids on closed-source apps. Accessibility needs to be more than a loosely sewn together illusion of usability. This kind of experience is not fair to those of us who are neurodivergent or who are afflicted with certain disabilities. If we want the web to be free and equal for all then we need to commit to making accessibility a core pillar of everything we create.

We hope Annota will fulfill the need for a virtual productivity environment designed to be as barrier-free as a web-based product can be. We believe that by being mindful of the interests of a growing and previously underserved [6] part of our community, we can tap into greater business potential, have an edge over existing competitors and lead the way for a more accessible future for all.

Conclusions and Recommendations for Future

7 Bibliography

[1] Team 6. Deliverable B: Needs identification and problem statement, GNG 2101B, 2020.

[2] Team 6. Deliverable C: Conceptual design, GNG 2101B, 2020.

[3] Team 6. Deliverable D: Detailed design, prototype 1, and BOM, GNG 2101B, 2020.

[4] Team 6. Deliverable G: Prototype 2, GNG 2101B, 2020.

[5] Team 6. Deliverable F: Business model, GNG 2101B, 2020.

[6] “Benefitting from accessible e-business practice”, *Ontario Government*, 11-Oct-2019.
[Online]. Available: <https://www.ontario.ca/page/benefiting-accessible-e-business-practices>
[Accessed 9-Dec-2020]

APPENDIX

The following is the list of the links presented in this user manual.

Maker Repo:

<https://makerepo.com/jodiqiao/annota-smart-annotation-compilation-b31>

GitHub:

<https://github.com/B31AccessibleAnnotations/B31-Annota>

Business Pitch:

https://www.youtube.com/watch?v=-gtoeTOzxD8&feature=youtu.be&ab_channel=TastyPastry

Video Demo:

https://www.youtube.com/watch?v=oJfJi673Pes&feature=youtu.be&ab_channel=RikkiRomana

Colour blind Google Chrome extension:

<https://chrome.google.com/webstore/detail/colorblinding/dgbgleaofjainknadoffbjkcllicbbgaa>