University of South Florida

ISM 6145 – Software Testing

Testing Portfolio Project

April 30, 2021

James Long

Contents

System Under Test and Testing Objectives

I chose Evernote as my system under test (SUT) for the portfolio project. The application capabilities center around a notebook and include a web clipper, handwriting recognition, document scanning, audio recordings, custom code blocks, team workspaces, single sign-on, and integration with other applications just to name a few. Three license levels are available including a basic level for free. The application is available as an app for Android, iOS, Windows, and Mac. There is also a web-based version supported by all major browsers. I tested using the basic license, the Windows app, the Android app, and the web-based client. The web-based client was required for Web Clipper, note sharing, attachments, Selenium, and structural testing analysis. Evernote recently introduced a new editor, which is the default in the web-based version, and new accounts cannot switch to the classic editor. However, the latest version of the Windows app does not yet support the new editor. On Android, either editor can be used. Similarly, support for other app features varies across the client types. Despite these variances, the overall user experience is quite similarly regardless of client type.

Some of the core notebook features include:

- Automatically saving notes online
- Automatically syncing notes across devices
- Offline note access
- Rich text and HTML formatting
- Support for images, PDFs, and Office documents in notes
- Checklists with completion tracking
- Tags
- Searching notes for titles, dates, content types, keywords, words in pictures and handwriting, and text in embedded PDF and Office files
- Text string encryption

The application has too many features to test every feature, so I elected to focus on a single feature for each of the five functional testing techniques required in this assignment. The features, testing techniques, and test objectives that I chose are:

- Encrypt Text in a Note – Function Testing – Deep dive on the capabilities of this feature
- Web Clipper – Exploratory Testing – Become familiar with this feature as it may be useful to me
- Cross-device Syncing – Stress Testing – Identify unreliable scenarios to explain issues I have experienced
- Note Sharing - Scenario Testing – Assess viability of this feature for my collaboration needs
- Attachments - Specification Testing – Identify any unsupported claims

Encrypt Text in a Note – Function Testing

Evernote supports encrypting simple text strings within notes. The feature is very valuable to users and yet easy to use due to its simplicity. However, it is somewhat limited. It cannot be used to encrypt non-text items such as images, text in tables, whole notes, or a notebook. Encrypted text is supported by the search engine via a special operator, "encryption:", which takes no arguments and finds all encrypted text strings. Encryption and decryption operations can be accessed via mouse or keyboard. Encrypted text can be decrypted temporarily for the current user need or permanently. Passphrases for encrypting text are never stored within Evernote locally or in the cloud, which makes the feature more secure but also introduces risk for users who do not manage their passphrases responsibly. Documentation for this feature is somewhat sparse and spread across different articles:

- https://help.evernote.com/hc/en-us/articles/209005547
- https://help.evernote.com/hc/en-us/articles/208313828
- https://help.evernote.com/hc/en-us/articles/209004807
- https://help.evernote.com/hc/en-us/articles/208314128

I tested the functionality and limitations for text encryption/decryption, non-text encryption/decryption (images, PDF files, Office files, non-ASCII characters), tables, passphrase reuse, multiple passphrases, search, copy/paste of encrypted text within Evernote, copy/paste of encrypted text outside of Evernote, export/import of encrypted text, printing of encrypted text, and deletion of encrypted text. Throughout testing, I used both mouse clicks and keyboard shortcuts to access the encryption feature. This test plan covered all functionality of the feature as far as I know.

Test Case	Description	Result	Test case report form
1	Encrypt single text string – do not remember	Text encrypted	
2	Temporarily decrypt single text string– do not remember	Text visible while viewing note, text not visible upon returning to note	
3	Permanently decrypt single text string– do not remember	Text visible while viewing note, text visible upon returning to note	
4	Encrypt note title	Not supported	Yes
5	Encrypt multiple text strings – do not remember	Text encrypted	Yes
6	Encrypt embedded jpeg file	Not supported	
7	Encrypt table containing only text	Partially supported	Yes
8	Encrypt embedded PDF or Office document containing only plain text	Not supported	
9	Encrypt separate text strings – remember	First string encrypted following prompt for passphrase, second string encrypted without prompt for passphrase, encrypting with a new passphrase requires an app restart	
10	Temporarily decrypt separate text strings – remember	Text visible while viewing note without prompt for passphrase, text not visible upon returning to note, decrypting with a different passphrase requires an app restart	

11	Permanently decrypt separate text strings – remember	Text visible while viewing note without prompt for passphrase, text visible upon returning to note, decrypting with a different passphrase requires an app restart	
12	Encrypt separate text strings – different passphrases	Received warning	Yes
13	Decrypt separate text strings – different passphrases	Text decrypted if correct passphrase is entered. Otherwise, the prompt for passphrase is redisplayed with a message indicating the wrong passphrase was entered. There appears to be no limit on the number of retries.	
14	Search for encrypted text using the special operator	All instances of encrypted text found in multiple notes	
15	Search for encrypted text using words that have been encrypted	No instances of encrypted words found in any note	
16	Copy encrypted text to another note then decrypt	Text copied and decrypted	Yes
17	Copy plain and encrypted text to another application	Plain text was copied. Encrypted text was not copied.	
18	Export note containing encrypted text	Encrypted text exported as an AES-128 encrypted string	
19	Import note containing encrypted text	All note content was displayed as it was before it was exported. Temporary and permanent decryption of strings worked properly using the previously associated passphrases.	
20	Print note containing encrypted text	Plain text was printed. Encrypted text was not printed.	
21	Encrypt non-ASCII characters	All characters encrypted and decrypted	
22	Delete encrypted text from note	Text deleted without prompting for passphrase	

FUNCTIONAL TEST CASE REPORTING FORMS

Test Case ID	4
Component	Encrypt Text in a Note
Purpose of Test Case	Determine if note title is treated differently by the encryption feature than note body
Functional Test Type	Function Testing
Pre-Conditions	PC running Windows 10 Evernote Windows app v10.4.4 build 55 public installed At least one notebook created containing at least one note with a title
Inputs	The note title as plain text

Expected Outputs	I expected to see the note title as an encrypted text string. Instead, the option to encrypt the note title was not present. I am guessing the reason is that the title serves as the primary identifier of each note. However, the app could allow one note title to be encrypted since this would not cause any ambiguity between notes. Furthermore, notes can be sorted, so sort order could be used to distinguish among multiple notes with encrypted titles. Of course, this has usability implications for infrequently accessed notes, notebooks with a large number of notes, and shared notes. The app producer apparently decided the pros did not outweigh the cons. This is likely the reason this feature does not support encryption of entire notes (i.e., the note title field must be included in the definition of a whole note). See test case 5.
Post-Conditions	Same as pre-conditions
Execution History	

Date	Result	Tester
2021-04-17	Pass	James Long

Test Case ID	5
Component	Encrypt Text in a Note
Purpose of Test Case	Determine if entire note body can be encrypted
Functional Test Type	Function Testing
Pre-Conditions	PC running Windows 10 Evernote Windows app v10.4.4 build 55 public installed At least one notebook created containing at least one note with a title and multiple lines of text in multiple paragraphs and empty lines between paragraphs
Inputs	Multiple lines of text in multiple paragraphs and empty lines between paragraphs
Expected Outputs	All text in note body encrypted. Note title not encrypted. Interesting observation: the note title cannot be selected with the note body.
Post-Conditions	Same as pre-conditions except text in note body is not visible
Execution History	

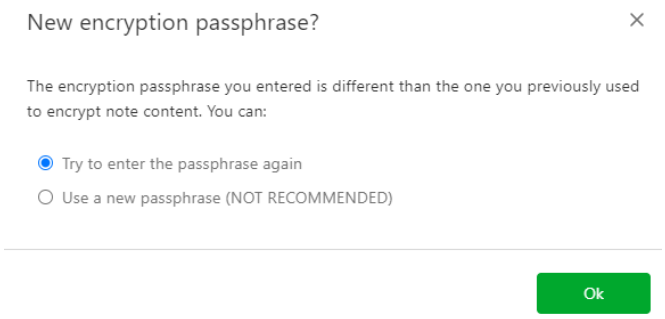
Date	Result	Tester
2021-04-17	Pass	James Long

Test Case ID	7
Component	Encrypt Text in a Note
Purpose of Test Case	Determine the limitations of table encryption
Functional Test Type	Function Testing
Pre-Conditions	PC running Windows 10 Evernote Windows app v10.4.4 build 55 public installed At least one notebook created containing at least one note with a multicell table containing only text
Inputs	A table containing only text A subset of cells within the table

	The text within a single cell in the table
Expected Outputs	I expected not to have the option of encrypting the whole table. This held true. I was not sure if text within a single cell would also be unsupported. However, selecting text within a single cell is supported and works exactly the same as text outside the table. Selecting text in more than one cell is not supported.
Post-Conditions	Same as pre-conditions except selected text within a single cell is not visible
Execution History	

Date	Result	Tester
2021-04-17	Pass	James Long

Test Case ID	12
Component	Encrypt Text in a Note
Purpose of Test Case	Determine behavior when using multiple passphrases
Functional Test Type	Function Testing
Pre-Conditions	PC running Windows 10 Evernote Windows app v10.4.4 build 55 public installed At least one notebook created containing at least one note with multiple text strings
Inputs	Two separate text strings, two different passphrases
Expected Outputs	I expected to be prompted for a passphrase for the second string and then to see the string displayed as encrypted text. Instead, I received a warning message before being allowed to proceed.



Upon choosing to use a new passphrase, I was allowed to enter the new passphrase.

When encrypting a third string with the original passphrase, I received the same warning. Upon cancelling and encrypting the third string with the second passphrase, I did not receive the warning. So, the program apparently has some form of passphrase retention during the current session even when the remember option is not selected. This was unexpected. I suspect the most recently used passphrase is cached in memory only since passphrases are not stored in Evernote.

The warning is useful considering all encrypted text appears the same despite the length or content of the strings. Additionally, encrypted strings can be dragged to different relative locations within a note, so recalling which passphrase to use for each string can be difficult.

Post-Conditions	Same as pre-conditions except for the presence of one or more encrypted strings

Execution History	

Date	Result	Tester
2021-04-17	Pass	James Long

Test Case ID	15
Component	Encrypt Text in a Note
Purpose of Test Case	Determine encryption capabilities when copying content between notes
Functional Test Type	Function Testing
Pre-Conditions	PC running Windows 10 Evernote Windows app v10.4.4 build 55 public installed At least one notebook created containing at least two notes
Inputs	A mix of plain text and encrypted strings in one note A separate note containing encrypted text A separate note containing no encrypted text
Expected Outputs	I expected copied plain text to be pasted as plain text. I expected copied encrypted text to be pasted as encrypted text with the original passphrase intact. This was the result. Decrypting copied text in the second and third notes had no impact on the same text in the first note and vice versa. Drag-and-drop between notes is not supported. To copy content, you must first select the content and then press control+c or use the copy option in the context menu by right clicking on the selected content. However, clicking on encrypted text does not select the text. And when right-clicking on encrypted text, the copy option is grayed out in the context menu. For this reason, I initially thought copy/paste of encrypted text was not supported. Then I entered a blank line above and below the encrypted text, so I could drag my mouse across the encrypted text. That made the copy option active on the context menu as long as I right-clicked one of the selected blank lines and not the encrypted text. This is not the most user-friendly way to facilitate copy/paste of encrypted text, but at least it is possible.
Post-Conditions	Same as pre-conditions plus additional encrypted text in the destination notes
Execution History	

Date	Result	Tester
2021-04-17	Pass	James Long

Web Clipper – Exploratory Testing

Evernote Web Clipper is a browser plugin that allows users to easily copy full or partial web pages into a note for subsequent viewing. All major browsers are supported, but the types of clips that are supported depend on the browser being used. I used Brave v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit). I have never used this feature, so exploratory testing seemed like the most appropriate testing technique. My objective was to learn about the feature's core value proposition while also discovering any limitations or defects. I started by reviewing the documentation:

https://help.evernote.com/hc/en-us/articles/209125877-Evernote-Web-Clipper-Quick-Start-Guide

The main clip types are simplified article, article, full page, bookmark, screenshot, and selection. Gmail, Amazon, LinkedIn, and YouTube also support specially formatted clips using checkboxes to select content. I mostly tested by browsing to CNN and Monster. CNN is a content rich site with complex page elements and heavy advertising, and it also provides the opportunity to test video playback within simplified article and article clips. By contrast, Monster has conventional blog entries that provide a simpler article page format for testing. I used the CNN homepage as a form of robust testing to see how simplified article and article clips would fare when capturing pages outside the scope of the intended use case. I used Amazon to explore the checkbox selection capability as well as the option to save an entire PDF as a note. To manage my time, I did not attempt to explore every capability of this expansive feature.

Note: I use Brave because it has very strong ad suppression and privacy features built in (called shields). However, these features can interfere with normal website operation is some cases. Likewise, these Brave features might interfere with Web Clipper, so I "lowered shields" in the Evernote tab and the other tabs involved in testing. I suspect anti-virus and personal firewall software could have a similar impact, though probably less dramatic. This has implications for compatibility and regression testing of Web Clipper.

Test Case	Description	Result	Test case report form
1	CNN article page as simplified article	Content captured as expected but with additional content	Yes
2	CNN video article page as simplified article	Video not captured	Yes
3	Monster article page as simplified article	Content captured as expected	
4	CNN home page as simplified article	Almost no content captured	Yes
5	CNN article page as article	Content captured as expected	
6	CNN video article page as article	Video not captured	Yes
7	Monster article page as article	Content captured as expected	
8	CNN home page as article	Almost no content captured	Yes
9	CNN article page as full page	Content captured as expected	
10	Monster article page as full page	Content captured as expected	
11	CNN article page as bookmark	Content captured as expected	
12	Monster article page as bookmark	Content captured as expected	
13	CNN article page as screenshot	Content captured and marked up as expected	
14	Monster article page as screenshot	Content captured and marked up as expected	

15	CNN article page as selection	Content captured as expected	
16	Monster article page as selection	Content captured as expected	
17	Amazon product page as checkbox selection	Content captured as expected	
18	Amazon product manual PDF saved as note	Content captured as expected, readable by paging through, original content preserved as text	

FUNCTIONAL TEST CASE REPORTING FORMS

Test Case ID	1
Component	Web Clipper
Purpose of Test Case	Explore capabilities and limitations when capturing a media rich article as a simplified article
Functional Test Type	Exploratory Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website and all test websites
Inputs	https://www.cnn.com/2021/04/24/politics/inequality-biden-100-days/index.html
Expected Outputs	The simplified article clip is supposed to remove excessive formatting, layouts, and unrelated content without excluding article content. I expected a simple list of sentences, links, and images. That is exactly what was captured in the top of the note. However, below the article, page after page of advertisements were captured with some images duplicated, many blank lines, a series of "untitled attachments", and privacy/cookie disclosures at the very bottom. Since this is all below the article content, it can easily be ignored. But I expected it to be eliminated from the capture.
Post-Conditions	If successful, same as pre-conditions plus a saved note containing only article content in simplified format. As tested, same as pre-conditions plus a saved note containing article content in simplified format plus excessive non-article content.
Execution History	

Date	Result	Tester
2021-04-17	Fail	James Long

Test Case ID	2
Component	Web Clipper
Purpose of Test Case	Explore capabilities and limitations when capturing a video article as a simplified article
Functional Test Type	Exploratory Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website and all test websites

Inputs	https://www.cnn.com/videos/world/2021/04/24/biden-armenian-genocide-ottoman-empire-johns-nr-vpx.cnn
Expected Outputs	I expected a working video link that would playback within the note. The video was not captured, and the top of the note contained excessive unrelated content in addition to the same unrelated content captured in test case 1.
Post-Conditions	If successful, same as pre-conditions plus a saved note containing only article content in a clickable, playable format. As tested, same as pre-conditions plus a saved note containing only a thumbnail image of the video plus excessive non-article content.
Execution History	

Date	Result	Tester
2021-04-17	Fail	James Long

Test Case ID	4
Component	Web Clipper
Purpose of Test Case	Explore capabilities and limitations when capturing a complex non-article page as a simplified article
Functional Test Type	Exploratory Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website and all test websites
Inputs	https://www.cnn.com/?refresh=1
Expected Outputs	I expected the content to be capture incorrectly but did not have specific expectations in mind. The only content saved was the title I gave to the note, the clip source link, a single line of content describing the site, and a line with copyright information. **CNN homepage as simplified article** **Clip source:** CNN homepage as simplified article Breaking News, Latest News and Videos © 2021 Cable News Network.A Warner Media Company.All Rights Reserved.CNN Sans ™ & © 2016 Cable News Network. This output might be the result of intelligence in the code that recognizes home pages and other highly complex content structures and extracts summary information from the page metadata. Since this test case was clearly outside the intended scope of the feature, I did not consider this a failure. In fact, the results are an elegant alternative to the jumbled mess that might otherwise have been captured.
Post-Conditions	Same as pre-conditions plus a saved note containing a brief site description and copyright info
Execution History	

Date	Result	Tester
2021-04-17	Pass	James Long

Test Case ID	6
Component	Web Clipper
Purpose of Test Case	Explore capabilities and limitations when capturing a video article as an article
Functional Test Type	Exploratory Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website and all test websites
Inputs	https://www.cnn.com/videos/world/2021/04/24/biden-armenian-genocide-ottoman-empire-johns-nr-vpx.cnn
Expected Outputs	I expected a working video link that would playback within the note. The video was not captured in playable format, but it was captured as an image that clearly indicated it is a video. Also, the article summary text was captured and displayed properly. All other content was captured and laid out as expected. Excessive non-article content was captured, but I expected that with the article clip type.
Post-Conditions	If successful, same as pre-conditions plus a saved note containing article content in a clickable, playable format plus other non-article content, all properly laid out. As tested, same as pre-conditions plus a saved note containing a non-playable video playback image plus other non-article content, all properly laid out.
Execution History	

Date	Result	Tester
2021-04-17	Fail	James Long

Test Case ID	8
Component	Web Clipper
Purpose of Test Case	Explore capabilities and limitations when capturing a complex non-article page as an article
Functional Test Type	Exploratory Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website and all test websites
Inputs	https://www.cnn.com/?refresh=1
Expected Outputs	I expected the content to be capture incorrectly but did not have specific expectations in mind. The only content saved was the title I gave to the note, a link to the source, and a box containing copyright information. This was even less content than was saved as a simplified article, which was unexpected. 

	This output might be the result of intelligence in the code that recognizes home pages and other highly complex content structures and extracts summary information from the page metadata. Since this test case was clearly outside the intended scope of the feature, I did not consider this a failure. In fact, the results are a somewhat elegant alternative (better if a brief site description was included) to the jumbled mess that might otherwise have been captured.
Post-Conditions	Same as pre-conditions plus a saved note containing site copyright info
Execution History	

Date	Result	Tester
2021-04-17	Pass	James Long

DEFECT REPORTING FORM

DEFECT ID: Evernote.WC.1

TITLE: Simplified Article Clip of Complex Article Page
REPORTED BY: James Long
REPORTED ON: 04/17/2021

COMPONENT: Evernote
SUBCOMPONENT: Web Clipper
VERSION: 7.13.8.31-f3c4893

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 1

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S3
CONSISTENCY: Regular

DEFECT SUMMARY:

When clipping a media rich article as a simplified article, excessive unrelated content is captured below the article content.

STEPS TO REPRODUCE:

Browse to any media rich article page such as https://www.cnn.com/2021/04/24/politics/inequality-biden-100-days/index.html
Click the Evernote icon in the browser
Click "Simplified article"
Click "Save clip"
View the note in Evernote Web

COMMENTS:

Non-article content should be filtered out by the simplified article clip type.

ATTACHMENTS:

1.

2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.WC.2

TITLE: Simplified Article Clip of Video Article Page
REPORTED BY: James Long
REPORTED ON: 04/17/2021

COMPONENT: Evernote
SUBCOMPONENT: Web Clipper
VERSION: 7.13.8.31-f3c4893

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 2

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When clipping a video article as a simplified article, excessive unrelated content is captured above and below the article content. Also, the article content (a video) is captured as a thumbnail image that is not linked to the video.

STEPS TO REPRODUCE:

Browse to any video article page such as https://www.cnn.com/videos/world/2021/04/24/biden-armenian-genocide-ottoman-empire-johns-nr-vpx.cnn
Click the Evernote icon in the browser
Click "Simplified article"
Click "Save clip"
View the note in Evernote Web

COMMENTS:

If video capture is not supported via the simplified article clip type, consider graying out this menu option when clipping video content.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.WC.3

TITLE: Article Clip of Video Article Page
REPORTED BY: James Long
REPORTED ON: 04/17/2021

COMPONENT: Evernote
SUBCOMPONENT: Web Clipper
VERSION: 7.13.8.31-f3c4893

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 6

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When clipping a video article as an article, the article content (a video) is captured as a playable video image that is not playable.

STEPS TO REPRODUCE:

Browse to any video article page such as https://www.cnn.com/videos/world/2021/04/24/biden-armenian-genocide-ottoman-empire-johns-nr-vpx.cnn
Click the Evernote icon in the browser
Click "Article"
Click "Save clip"
View the note in Evernote Web

COMMENTS:

If video capture is not supported via the article clip type, consider graying out this menu option when clipping video content.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

Multi-Device Syncing – Stress Testing

One of the main benefits of Evernote is having access to your notes from anywhere, anytime, and from multiple devices. To facilitate this benefit, Evernote must make your data accessible from a central point. Thus, Evernote stores all notes in the cloud by default. I did not expect to encounter any defects with this testing because this capability is central to Evernote's value proposition. Any issues with syncing across devices could severely damage their reputation and even put them out of business.

https://help.evernote.com/hc/en-us/sections/115004021767-Syncing

Notes created via the app are created locally and then synced to the cloud before being synced to any other device. Since notes can be created on multiple devices at the same time, conflicts can occur. Evernote detects these conflicts to prevent data loss. Conflicting notes are placed in a special notebook named Conflicting Changes that Evernote creates dynamically. The user can then manually review the notes to decide what content to keep.

With the basic license, only two devices are allowed. Thus, introducing a third device forces the user to deactivate an existing device. This could create syncing issues if the timing occurs during a sync activity. With either of the advanced licenses, this test would be harder to simulate because they both support unlimited devices. That means adding a new device would simply increase the number of destinations.

As with all online services, there are limits on the amount of bandwidth the servers can allocate to any given user at any given time, and these limits can be imposed in several ways. Transmission throttling is one way, and it could conceivably interfere with the sync process. The Basic license imposes a 25MB maximum note size and a 60MB monthly upload limit. However, Evernote does not disclose the details of how the upload limit is imposed. So, I tested using the largest attachment size and a fast network connection (35Mbps upload) to see if transmission throttling might interfere with syncing. I took both devices offline, attach a 25MB file to a uniquely named note on each device, and then placed both devices back online simultaneously. This test would be more effective with a much larger file. However, the advanced licenses are required for larger file sizes, and both licenses increase monthly upload limits (10GB and 20GB). Thus, throttling is less likely with those licenses. A faster upload speed would also improve the tester's ability to stress the servers, but I have limited access to other (faster) networks. So, I tested during business hours hoping to catch the servers during peak load.

Intermittent connectivity is common when traveling and using a mobile device. I simulated this by using my Wi-Fi router to intermittently disconnect/reconnect my test devices while syncing multiple notes. Similarly, low bandwidth and high latency connections could interfere with syncing. I simulated low bandwidth connections by using my Wi-Fi router to throttle the bandwidth of my test devices.

Surprisingly, the Evernote website mentions that notes in the "Trash" notebook can sometimes cause sync problems. I suspect this is a rare event for the reasons previously mentioned. They advise emptying the trash if sync issues occur. So, I attempted to sync notes with a large number of deleted notes in my Trash notebook.

To round out the testing, I simply exceeded the published limits for the Basic license. I did not expect any errors or loss of data. However, I did expect some type of message to the user and local preservation of notes.

Test Case	Description	Result	Test case report form
1	Existing synced note edited on different devices while offline to create a sync conflict	Conflict detected, and no data lost	Yes
2	New note with same name created on different devices while offline to create a sync conflict	Conflict detected, and no data lost	
3	Third device connected to account while note sync in process	Device change detected and treated as a conflict, but no data lost	Yes

4	Sync two large attachments in opposite directions at same time	Notes synced without issue	
5	Sync multiple notes while simulating intermittent connectivity failures	Notes synced without issue	Yes
6	Sync multiple notes while simulating low bandwidth connectivity	Notes synced without issue, took a long time but did not error out	
7	Sync multiple notes while Trash notebook full of deleted notes	Notes synced without issue	
8	Sync more than the monthly upload limit	Warning messages displayed as expected, local preservation did not occur	Yes
9	Sync a note in excess of maximum note size	"Note size over limit" message displayed as expected, note not saved locally when created with a single large file, note saved locally and synced with partial content when created with a collection of medium sized files	

FUNCTIONAL TEST CASE REPORTING FORMS

Test Case ID	1
Component	Note Sync
Purpose of Test Case	Test note conflict behavior with an existing note
Functional Test Type	Stress Testing
Pre-Conditions	PC running Windows 10 Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on the PC Evernote app for Android v8.12.4 installed on another device One note created with any text and synced to both devices Both devices taken offline
Inputs	Any text entered into the pre-existing note on each device Connectivity restored to both devices
Expected Outputs	I expected Evernote to detect the conflict, dynamically create a notebook named Conflicting Changes (per the documentation), and then store the conflicting notes in the new notebook for manual review. Evernote did detect the conflict and displayed a popup warning in the Windows app. However, the Android app did not display a warning. This could lead to some confusion for the user. Both apps displayed both versions of the note in my default notebook. A temporary notebook named Conflicting Changes was not created. However, no data was lost, and manually consolidating the notes was easy and intuitive.
Post-Conditions	If successful, same as pre-conditions plus an additional note by the same name on each device.
Execution History	

Date	Result	Tester
2021-04-18	Pass	James Long

Test Case ID	3
Component	Note Sync
Purpose of Test Case	Test behavior when a device change is made to the account while a sync activity is in progress
Functional Test Type	Stress Testing
Pre-Conditions	Two PCs running Windows 10 Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on the PCs Evernote app for Android v8.12.4 installed on another device One note created with large content on the Android device Android device taken offline
Inputs	Connectivity restored to the Android device A new device associated with the account immediately after the Android device comes online
Expected Outputs	I was not sure what to expect. On the device being added, Evernote displayed a message stating the device limit had been exceeded and prompted me to upgrade my license or select a device to disconnect. I selected the other Windows device so the source of the syncing content (Android device) could continue transmitting. All content from the cloud was synced to the new device as was the large note being actively synced from the Android device. However, the large note was renamed with "Note conflict" prepended to the note name. This new name appeared on the Android device too. On the new device (Windows), a warning was displayed indicating a note conflict occurred. It was the same warning displayed in test cases 1 and 2. Unlike test cases 1 and 2, a warning was displayed on the Android device. However, it was different than the Windows warning and also contained a timestamp that the Windows warning omitted. Only one instance of each note appeared on the new device, so no conflict actually occurred. It seems the developers chose to reuse note conflict code as a way of informing the user that something went awry during the sync process. However, the warnings and renaming of the note are misleading since no data was lost, and no duplicate notes were created. On the other Windows device, Evernote displayed the same message about the device limit being exceeded. After I deactivated this device on the other Windows device, this device was automatically logged out of the account with no further messages. Overall, the process proved to be quite reliable, so I do not consider this a failure. Nonetheless, the user experience could be improved to reduce confusion and avoid unnecessarily alarming the user.
Post-Conditions	If successful, an existing device will be disassociated from the account, a new device will be associated with and logged into the account, all content from the cloud and from the third device will be synced to the new device without loss of data, and the disassociated device will be logged out from the account.
Execution History	

Date	Result	Tester
2021-04-18	Pass	James Long

Test Case ID	5
Component	Note Sync

Purpose of Test Case	Test sync behavior in the presence of intermittent connectivity failures
Functional Test Type	Stress Testing
Pre-Conditions	PC running Windows 10 Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on the PC Evernote app for Android v8.12.4 installed on another device One device (the source) taken offline Multiple notes created with any text on the source device while offline
Inputs	Connectivity restored to source device to begin sync process Intermittent loss of connectivity introduced to the source device while sync is in progress
Expected Outputs	I expected Evernote to halt the sync process when connectivity was lost and automatically restart it when connectivity was restored. This is what happened. I also expected some notes to be fully synced while connectivity was present and others in flight to be cancelled with no partially filled notes appearing on the destination device. This too is what happened. What was unexpected was the lack of warnings or other notifications from Evernote, especially after test case 3. Evernote did not display any messages during the process. I only saw a message when I attempted to manually sync while offline, and Android generated that message. Overall, the process proved reliable although a bit "quiet".
Post-Conditions	If successful, same as pre-conditions plus all notes synced on both devices
Execution History	

Date	Result	Tester
2021-04-18	Pass	James Long

Test Case ID	8
Component	Note Sync
Purpose of Test Case	Test sync behavior in excess of monthly upload limit
Functional Test Type	Stress Testing
Pre-Conditions	PC running Windows 10 Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on the PC
Inputs	A series of notes large enough to exceed the monthly upload limit
Expected Outputs	A warning message was displayed in a yellow box at 75% of the monthly limit. Another warning message was displayed in a red box at 95% of the monthly limit. Upon exceeding the monthly limit, a small popup window displayed a message that the limit has been exceeded. It prompted me to either return to my note or upgrade my license. Upon returning to my note, the newly added content was gone. This was unexpected, but after I thought about it, it made sense. The sync process is automatic. If the excess content were allowed to remain, the sync process would continually attempt to upload it, resulting in a perpetual series of popup messages to the user.
Post-Conditions	Same as pre-conditions minus the ability to create or modify notes until the next month.
Execution History	

Date	Result	Tester
2021-04-18	Pass	James Long

Note Sharing – Scenario Testing

Notes in Evernote are most akin to documents. In most professional settings, document collaboration needs are met by enterprise-grade solutions from Microsoft, Apple, Google, or The Document Foundation. While Evernote offers a Business license, this really stretches my imagination considering the alternatives. By contrast, Evernote's note sharing feature clearly has potential for personal use. Since scenario testing is supposed to focus on a persuasive story that reflects real world usage, I opted for a personal use scenario that could be beneficial to me: sharing a shopping list with my family members.

Evernote provides three ways to share a note. First, a note can be shared via email. This requires an advanced license, and only a snapshot of the note is shared. Changes to the note are not updated in the shared snapshot. Next, a note can be shared via a public link. When the link is enabled, anyone with the link can see (but not edit) the note. Security is a major concern with this method of sharing. This is useful for "publishing" information such as a meeting place/time for social gatherings or professional networking events, but it is not very useful for collaboration.

https://help.evernote.com/hc/en-us/sections/115004021787-Sharing

The third option is of value to me personally as it facilitates note collaboration. With this option, a note is shared within Evernote by selecting the email address of one or more other Evernote account holders, setting the permissions level for each recipient, and then sending a notification. I use Evernote for shopping lists among other things. I can see this feature making my life easier by sharing shopping list updates with my family members in real-time.

https://help.evernote.com/hc/en-us/articles/209005417

In the following scenario, I create a shopping list and then share it with a family member. That family member then edits the shopping list. Then she shares it with another family member. The idea is to provide some flexible among trusted collaborators to share as appropriate for ongoing collaboration. Unfortunately, the scenario failed at this point, and another tact was required. Consequently, some test cases appear that are not normal steps in the scenario. The required adjustment represents a minor limitation to our ability to share notes flexibly, so the remaining steps in the scenario are still viable. Next, each person makes edits to the note in sequence. In the final step, edits are made by more than one person at the same time. At this point, the scenario failed again in a way that makes the solution non-viable for valuable data or large groups of people but still acceptable for shopping list data among a few family members.

Test Case	Description	Result	Test case report form
1	Create a test note for sharing	Note created	
2	Share note with another Evernote account, grant full permissions (view, edit, share)	Note shared	Yes
3	Assess viewability of shared notes on recipient device	Shared note not listed in "All Notes", only listed in "Shared with me", avoids mixing personal and shared notes	
4	Edit shared note in recipient account	Edit made without issue, changes displayed in source account	
5	Share note from recipient account with third account (view, edit, share)	Note shared	Yes
6	Verify permissions of each account	Many issues identified	Yes
7	Unshare note from both recipient accounts	Access removed, but visibility not removed	Yes

8	Share note directly to two other accounts from source account (view, edit)	Note shared	
9	Verify permissions of each account	Security improved satisfactorily	Yes
10	Edit shared note on source device (text)	Edits updated on other devices	
11	Edit shared note on second device (text)	Edits updated on other devices	
12	Edit shared note on third device (photo of product label)	Edits updated on other devices	
13	Edit shared note on multiple devices simultaneously	Conflicts managed, and no data actually lost, but conflicts are not visible, so data is effectively lost	Yes

FUNCTIONAL TEST CASE REPORTING FORMS

Test Case ID	2
Component	Share a note within Evernote
Purpose of Test Case	Determine behavior when sharing a note
Functional Test Type	Scenario Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website Evernote app for Android v8.12.4 installed on another device and logged into a different account A note created in one account
Inputs	The note The email address associated with the other Evernote account
Expected Outputs	I expected some indication in the source account that the note had been shared. Indeed, an icon was displayed on the note that intuitively conveyed the note was shared. I also expected the email received by the other account holder to provide some information about the note. While no specific information was included (e.g., note name, etc.), a hyperlink was provided. Clicking the link opened a prompt in my browser to continue with the Evernote Web or the Windows App. That was a nice touch. Presumably, on an Android device, the prompt would have mentioned Evernote Web or the Android App. One odd thing was the wording in the email. It stated, "<source account> is using chat to share work with you." Evernote is not a chat application, and I declared my primary use case to be "Student" when I created each test account. So, I was not sure why they chose that wording. Some research revealed they are working on a new "work chat" feature. Still, the wording is misleading for Basic license holders using the app for non-work reasons. On the device logged into the recipient account, I was required to verify my email account before I could see shared notes. This was necessary because the account creation process did not require email verification. While this could be construed a defect, it relates to account creation and not to note sharing, so I will not include a defect report for this. After verifying my email address, the note was accessible in the

	"Shared with me" notebook. Having all shared notes appear in a specially named, dedicated notebook makes it easy to locate them. I also saw a temporary popup message in the app alerting me to the new note. That was another nice touch in case I had not checked email before using the app. I did not need to "accept" the invitation to gain access to the note; the email and popup were merely for awareness. The newly shared note displayed the email address of the sharing account and a timestamp.
Post-Conditions	Same as pre-conditions except the note is now shared between two accounts
Execution History	

Date	Result	Tester
2021-04-24	Pass	James Long

Test Case ID	5
Component	Share a note within Evernote
Purpose of Test Case	Determine behavior when sharing an already shared note
Functional Test Type	Scenario Testing
Pre-Conditions	Two PCs running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed on one PC Brave shields down for Evernote website Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on one PC Each PC logged into a different Evernote account Evernote app for Android v8.12.4 installed on another device and logged into a different account A note created in one Windows account and shared with the Android account Classic editor selected on all clients
Inputs	The note The email address associated with the second Windows account
Expected Outputs	I expected a very similar experience to test case 2. The only difference was that the Android app allowed me to include a personal message when sharing the note. That would be nice in the Windows app too. I also expected the original account to be notified of the sharing activity. To accomplish that, the icon was updated with a number indicating the total accounts with note access (3).
Post-Conditions	Same as pre-conditions except the note is now shared between three accounts
Execution History	

Date	Result	Tester
2021-04-24	Pass	James Long

Test Case ID	6
Component	Share a note within Evernote
Purpose of Test Case	Verify note-level permissions granted to each account
Functional Test Type	Scenario Testing

Pre-Conditions	Two PCs running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed on one PC Brave shields down for Evernote website Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on one PC Each PC logged into a different Evernote account Evernote app for Android v8.12.4 installed on another device and logged into a different account A note created in one Windows account, shared with the Android account, and shared by the Android account with the second Windows account
Inputs	No input was required for this test case since I was merely verifying the state of the note permissions within each account.
Expected Outputs	On the source device, I could modify or revoke access for each of the other accounts. This was expected. However, I could not see who the third account holder was (no name or email address was displayed). This is a defect and suggests the best practice is to share only from the source account while preventing other accounts from sharing. From the second account (Android), I could only modify or revoke access for the third account. This was reasonable and expected. I could also see the account info for all three accounts. This makes sense since both of the other accounts have a direct relationship to the Android account. Interestingly, I could not modify or revoke my own access to the note. Moreover, I could not delete the note, and I was not presented with a prompt to accept the shared note. So, sharing a note with someone effectively forces the recipient to keep the note. This could be used as a form a denial-of-service (DoS) attack. I found an Evernote article that addresses this concern for shared notebooks, but some ambiguity remains regarding how upload data is counted for shared notes. If the code fails to account for upload data of shared notes in the same way as shared notebooks, this is a defect. For example, I could share a very large note and make edits to it until I consume the allotted monthly upload capacity of the recipient account. The recipient would no longer be able to create or edit his/her own notes. I did not fully test this potential exploit because I wanted to stay focused on my scenario. However, I did produce a defect report since this has security implications. Moving on, I was able to email a copy of the note to anybody else. While sharing via email is supposedly only available via advance license, this appears to be a way around the license constraint (but only for shared notes). This is yet another defect since it can be used to expose private data unbeknownst to the other shared account holders. Finally, I was able to turn on the public link sharing feature for the note. This represents yet another security risk and thus a defect since the source account was not afforded the opportunity to approve/deny the action. The note status was updated in the other accounts, so at least the other account holders had a way of discovering the exposure. However, no alert or message was displayed in the other accounts, so those account holders would need to frequently check the status of the note to detect the exposure. Clearly, that is not reasonable or practical. From the third account (Windows), I noticed several more issues, but I decided not to document these. Since so many defects were already discovered from the first and second account perspectives, this configuration option is not viable. If note sharing is to be used securely, the source account must not allow other accounts to add new accounts to the note.
Post-Conditions	Same as pre-conditions
Execution History	

Date	Result	Tester
2021-04-24	Fail	James Long

Test Case ID	7
Component	Share a note within Evernote
Purpose of Test Case	Verify removal of shared notes denies further access
Functional Test Type	Scenario Testing
Pre-Conditions	Two PCs running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed on one PC Brave shields down for Evernote website Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on one PC Each PC logged into a different Evernote account Evernote app for Android v8.12.4 installed on another device and logged into a different account A note created in one Windows account, shared with the Android account, and shared by the Android account with the second Windows account
Inputs	Remove access to the shared note
Expected Outputs	On the source device, I was able to revoke access to the note for both of the other accounts. However, the note was still visible in the "Shared with me" notebook on both of the other devices. I logged out and cleared local cache on both devices, but upon logging back in, the note was still visible on both devices. Upon clicking the note on either device, a message displayed stating the note is no longer accessible. But even that did not clear the note from the notebook. There appears to be no way to remove the ghost image of the now inaccessible note.
Post-Conditions	If successful, same as pre-conditions except the shared note is no longer accessible or visible on either recipient device As tested, same as pre-conditions except the shared note is no longer accessible but is still visible on the recipient devices
Execution History	

Date	Result	Tester
2021-04-24	Fail	James Long

Test Case ID	9
Component	Share a note within Evernote
Purpose of Test Case	Verify note-level permissions granted to each account
Functional Test Type	Scenario Testing
Pre-Conditions	Two PCs running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed on one PC Brave shields down for Evernote website Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on one PC Each PC logged into a different Evernote account Evernote app for Android v8.12.4 installed on another device and logged into a different account A note created in one Windows account and shared directly with other accounts

Inputs	none
Expected Outputs	No input was required for this test case since I was merely verifying the state of the note permissions within each account. On the source device (web client), I could see the email address but not the name of the Android account. I could see the name and email address of the other Windows account (Windows app). I could modify or revoke access for each of the other accounts. This was expected. I could also see the account info for both other accounts (name and email address). This too was expected. The options to share with additional account holders or via public link were available as expected. Clicking "email a copy" prompted me to upgrade my license. On the Android device, I could see the name but not the email address of the source account. I could see the name and email address of the other Windows account (Windows app). I could not edit the access of the other accounts or my own account, nor could I add additional accounts. I could not share via email or public link. On the second Windows device (Windows app), I could see the name but not the email address of the other accounts. I could not edit the access of the other accounts or my own, nor could I add additional accounts. I could not share via email or public link. The inconsistencies in account metadata visibility are a minor annoyance but not a security concern. All security concerns were eliminated by sharing with this method.
Post-Conditions	
Execution History	

Date	Result	Tester
2021-04-24	Pass	James Long

Test Case ID	13
Component	Share a note within Evernote
Purpose of Test Case	Verify note conflicts are managed well for shared notes
Functional Test Type	Scenario Testing
Pre-Conditions	Two PCs running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed on one PC Brave shields down for Evernote website Evernote app for Windows v10.12.5-win-ddl-public (2564) installed on one PC Each PC logged into a different Evernote account Evernote app for Android v8.12.4 installed on another device and logged into a different account A note created in one Windows account and shared directly with other accounts
Inputs	Any edit made to the shared note on two or more devices at the same time
Expected Outputs	When one device edits the note, an alert is displayed on the other devices in near real-time indicating the note is being edited by another device. Attempts to edit the note on the other devices are denied until the first device exits edit mode and updates are synced to all devices. This is a good solution, but it does not always work as expected. A lag occurs from the start of editing to the alert displaying due to transmission delay. So, multiple devices can start editing the note at the same time. When this happens, a note

	conflict is detected. However, the saved copies of the conflicting notes are not shared to all accounts. They are only visible to the source account. No alerts or email notifications are sent to any account indicating a conflict occurred. Thus, the conflict goes undetected by all users except the source account user. While no data is actually lost, data seems to be lost from the perspective of all non-source account users. The more people who share the note, the worse the impact because note conflicts will occur with greater frequency.
Post-Conditions	Same as preconditions plus one or more conflicting notes visible only to the source account
Execution History	

Date	Result	Tester
2021-04-24	Fail	James Long

DEFECT REPORTING FORM

DEFECT ID: Evernote.Sharing.1

TITLE: Cannot See Downstream Account Info
REPORTED BY: James Long
REPORTED ON: 04/24/2021

COMPONENT: Evernote
SUBCOMPONENT: Note Sharing
VERSION: v10.12.5-win-ddl-public (2564)

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 6

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When a note is shared with another account, and then that account holder shares the note with a third account, the third account's information is not visible to the first account holder.

STEPS TO REPRODUCE:

Create a note
Share the note with another account
From the second account, share the note with a third account
From the first account, view the sharing status of the note

COMMENTS:

The source account should have full visibility to all other accounts with access to the note.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Sharing.2

TITLE: DoS Attack by Sharing a Large Note
REPORTED BY: James Long
REPORTED ON: 04/24/2021

COMPONENT: Evernote
SUBCOMPONENT: Note Sharing
VERSION: v10.12.5-win-ddl-public (2564)

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 6

RESOLUTION PRIORITY: P1
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

The app does not crash, but access could be denied, which is on par with a crash. So, I consider this a P1. Sharing a note with another account does not require the recipient account to accept the note. Thus, a large note could be shared and repeatedly edited until the monthly upload limit of the recipient account is exceeded. Alternatively, a large note could be shared and unshared repeatedly. Either scenario could deny the account holder access to edit his/her other notes. While this is more easily accomplished against a Basic license account, it can be accomplished against advanced license accounts too with the use of larger notes.

STEPS TO REPRODUCE:

Create a large note (relative to the license upload limit)
Share the note with another account
Make substantive edits to the note content repeatedly until the monthly upload limit is reached
Alternatively, unshare and reshare the note repeatedly until the monthly upload limit is reached

COMMENTS:

This defect report is unverified and arises from an ambiguity in the documentation. If this exploit is real, shared note upload data should be counted toward the note owner's account in the same way shared notebooks are, or shared notes should require acceptance by the recipient account.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Sharing.3

TITLE: Note exposure via email
REPORTED BY: James Long
REPORTED ON: 04/24/2021

COMPONENT: Evernote
SUBCOMPONENT: Note Sharing
VERSION: v10.12.5-win-ddl-public (2564)

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 6

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When a note is shared with another account, the recipient account is able to send a copy of the note via email to any valid email address. The source account is not notified and cannot prevent the email from being sent.

STEPS TO REPRODUCE:

Create a note
Share the note with another account
From the recipient account, access the note
Click the share icon
Click the menu icon
Click "email copy of note"

COMMENTS:

Sharing notes via email is supposed to be restricted to Premium and Business licenses. Either the documentation is wrong, or the code has a defect.

If a shared note is sent via email to any recipient that does not already have access to the note within Evernote, the source account should be given notice and have the ability to approve or deny the sharing attempt before the note is shared.

There may also be further restrictions needed for shared notes (e.g., preventing content from being copied into the clipboard).

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Sharing.4

TITLE: Note exposure via public link
REPORTED BY: James Long
REPORTED ON: 04/24/2021

COMPONENT: Evernote
SUBCOMPONENT: Note Sharing
VERSION: v10.12.5-win-ddl-public (2564)

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 6

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When a note is shared with another account, the recipient account is able to expose the note via public link. The source account is not notified and cannot prevent the exposure.

STEPS TO REPRODUCE:

Create a note
Share the note with another account
From the recipient account, access the note
Click the share icon
Click "Shareable link off" to toggle the link on

COMMENTS:

If a shared note is to be exposed via public link, the source account should be given notice and have the ability to approve or deny the exposure.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Sharing.5

TITLE: Shared note visibility not removed
REPORTED BY: James Long
REPORTED ON: 04/24/2021

COMPONENT: Evernote
SUBCOMPONENT: Note Sharing
VERSION: v10.12.5-win-ddl-public (2564)

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 7

RESOLUTION PRIORITY: P3
RESOLUTION SEVERITY: S2
CONSISTENCY: Regular

DEFECT SUMMARY:

When a note is unshared, the recipient account(s) can still see the note. Access to edit the note is denied.

STEPS TO REPRODUCE:

On the source device, revoke access to a shared note by all other accounts
From the recipient account, access the "Shared with me" notebook
The formerly shared note will be visible

COMMENTS:

When a note is unshared, both access and visibility should be removed from the recipient account.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Sharing.6

TITLE: Note Conflicts not Shared
REPORTED BY: James Long
REPORTED ON: 04/24/2021

COMPONENT: Evernote
SUBCOMPONENT: Note Sharing
VERSION: v10.12.5-win-ddl-public (2564)

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 13

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When a note is shared by multiple accounts, and a note conflict occurs, the conflicting notes are not shared to all accounts. From the perspective of all but the source account (i.e., the account that shared the note), this appears as a loss of data.

STEPS TO REPRODUCE:

Share a note from one account to multiple other accounts
Edit the note from multiple accounts simultaneously
View the "Shared with me" notebook from all non-source accounts
View the "Notes" menu from the source account

COMMENTS:

When a note conflict occurs on a shared note, the conflicting copies of the note should be shared to all the accounts that have access to the original note.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

Attachments – Specification Testing

Notes that do not support file attachments are of limited value. So, it is no surprise that Evernote supports a variety of attachment formats. Furthermore, Evernote integrates with numerous other apps and services such as Skitch, Slack, Google Drive, Gmail, SMTP forwarding, and more. Each of these integrations can be used to attach files to a note in Evernote. To test all claims made with respect to file attachments via these third-party apps and services would be quite time consuming. So, I limited the scope of my attachment testing to native app capabilities.

The native attachment capabilities vary somewhat depending on the Evernote app being used. Also, Evernote introduced a new editor recently that will be the only choice eventually. But for now, the new editor has some known limitations relative to the classic editor (some of which are attachment oriented). For these reasons, I chose to test in the web client and the Android app using the classic editor in each. Native attachment capabilities also vary by license. Some features, such as business card scanning, could not be tested due to license constraints.

To develop my test plan, I first needed to identify the attachment-related claims made in Evernote's documentation and marketing materials. This task is never easy because claims can be spread across a multitude of sources. Nonetheless, the following table summarizes the relevant claims I found and associates each with one or both tested apps.

Claim ID	App	Claim Details	URL
1	Web	Any type of file can be attached to a note, as long as the total size of the note with attachments included does not exceed the note size limit of your account.	https://help.evernote.com/hc/en-us/articles/208313688
2	Web	Drag and drop the file from your computer onto the note body.	https://help.evernote.com/hc/en-us/articles/208313688
3	Web	Click on the blue plus sign icon in the note editor and select Attachment to manually attach a file.	https://help.evernote.com/hc/en-us/articles/208313688
4	Android	Any type of file can be attached to a note, as long as the total size of the note with attachments included does not exceed the note size limit of your account.	https://help.evernote.com/hc/en-us/articles/208313688
5	Android	Tap the arrow on the right side of the button to access additional quick note options. Start sketching: Immediately open the sketch window in a new note.	https://help.evernote.com/hc/en-us/articles/219925607
6	Android	Use the Evernote camera to snap a photo of anything. Capture handwritten notes and drawings from a whiteboard or the back of an envelope.	https://help.evernote.com/hc/en-us/articles/222177927
7	Android	Scan paper documents, business cards, product labels, and manuals. Handwritten or printed text saved to Evernote becomes searchable. Clearly written notes are editable.	https://help.evernote.com/hc/en-us/articles/222177927
8	Android	You can record and store an audio recording directly into Evernote. Tap on the blue plus sign icon, then tap Audio. Recording starts automatically. When you're done,	https://help.evernote.com/hc/en-us/articles/208314418

		tap the stop icon to stop recording and save the audio to your note.	
9	Android	You can type inside Evernote as audio is recording.	https://help.evernote.com/hc/en-us/articles/208314418
10	Android	If you have a pre-recorded audio file, you can drag the file right into Evernote to create a new note.	https://help.evernote.com/hc/en-us/articles/208314418
11	Android	Most common audio formats are supported, including .mp3, .mp4, .flac, and .wav.	https://help.evernote.com/hc/en-us/articles/208314418
12	Web, Android	If a note contains a file (like a PDF, image or other file) and you'd like to save a copy of this file to your computer for use in other applications, simple right-click the file and choose "Save As..." and choose the location where you'd like to save the file. You may also want to change the filename since, depending upon how the file was added, the filename may be a series of letters and numbers generated by the Evernote application when the file was added.	https://help.evernote.com/hc/en-us/articles/209005097
13	Web, Android	You can search for notes and notebooks by: Text in attached documents and PDFs	https://help.evernote.com/hc/en-us/articles/209005647
14	Web, Android	What features are not yet available in the new editor? Non-PDF file attachment previews	https://help.evernote.com/hc/en-us/articles/360022954093

Test Case	Description	Result	Test case report form
1	Claim 1	Files were attached	Yes
2	Claim 2	Files were attached	Yes
3	Claim 3	Simultaneously tested with claim 1, no issues	
4	Claim 4	Files were attached	
5	Claim 5	Sketch file created within a new note. This could be considered a test of the sketch feature, but since the final result is a note with an attachment, it can be viewed as an alternative method of attaching a file to a note.	
6	Claim 6	Files attached as images. Text in images was as legible as the source documents.	
7	Claim 7	Not supported	Yes
8	Claim 8	Audio file created within a new note. This could be considered a test of the audio recording feature, but since the final result is a note with an attachment, it can be viewed as an alternative method of attaching a file to a note.	
9	Claim 9	Able to type into new note as recording was occurring	

10	Claim 10	Able to create a new note by drag-and-drop of audio file	
11	Claim 11	All stated file formats can be attached to notes	
12	Claim 12 in web client	Claim not supported exactly as stated, but an alternative method exists with trivial limitations. Claim fails under rare circumstances.	Yes
13	Claim 12 in Android	Claim not supported exactly as stated, but an alternative method exists with no limitations.	
14	Claim 13 in web client	Claim partially supported	Yes
15	Claim 13 in Android	Claim not supported in Basic license. Popup message prompted me to upgrade to be able to search within documents. This is a documentation defect.	
16	Claim 14 in web client	Preview worked for PDF file, and the file could be paged through by clicking arrows. Preview also worked for .txt, .jpg, and .png files.	
17	Claim 14 in Android	Preview worked for PDF file, and the file could be scrolled through by swiping up/down. Preview also worked for .jpg and .png files but not for .txt files.	

FUNCTIONAL TEST CASE REPORTING FORMS

Test Case ID	1
Component	Attachments
Purpose of Test Case	Verify claim that any type of file can be attached to a note
Functional Test Type	Specification Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website A note created
Inputs	Files of the following types: .whl, .txt, .exe, .msi, .bat, .py, .html, .htaccess, .zip, .mhtml, .jpg, .png, .pdf, .wbk, .docx, .pptx, .odp, .odt, .ods, .xlsx, .json, .rtf, .mp3, .m4a, .mp4, .enex I tested with real data. In other words, I did not test with a single file by changing the file extension. This is germane because another claim by Evernote is that certain attachment contents can be searched. Thus, the app must be able to open certain types of attachments and read the contents. That implies the app must be able to gracefully handle unreadable content. I could have tested with additional file types (e.g., SQLite file, etc.), but my test was sufficiently varied to confidently state this claim is supported under normal usage. The .enex file type is an exported Evernote note. I was not sure if it would be attached as a file or recognized as a note resulting in the creation of a new note. The file was attached to the test note.

	The first time I attempted to attach a PDF file, the app displayed an error temporarily, "Something went wrong with your attachment. Please try again." I was unable to reproduce the error, so this appears to have been a transient server issue.
Expected Outputs	Files attached without error.
Post-Conditions	Same as pre-conditions except the note will have many attachments
Execution History	

Date	Result	Tester
2021-04-25	Pass	James Long

Test Case ID	2
Component	Attachments
Purpose of Test Case	Verify claim that files can be attached to notes via drag-and-drop onto the note body
Functional Test Type	Specification Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website A note created
Inputs	Any file
Expected Outputs	I expected the file to be attached to the note. This happened as claimed. Interesting, when I dragged a file onto the note name in the Notes list, I was prompted to create a new note. Evernote does not claim this will work, so I did not consider it a defect. But intuitively, this should attach the file to the note. Dragging onto empty space in the Notes list should prompt to create a new note. Dragging onto any other part of the app had no effect.
Post-Conditions	Same as pre-conditions except a file is attached to the note
Execution History	

Date	Result	Tester
2021-04-25	Pass	James Long

Test Case ID	7
Component	Attachments
Purpose of Test Case	Verify claim that scanned attachments containing clearly handwritten text can be searched and edited
Functional Test Type	Specification Testing
Pre-Conditions	Evernote app for Android v8.12.4 installed
Inputs	A clearly handwritten note
Expected Outputs	I expected a new note containing an image of the "scanned" document, and that is what I got. Interestingly, the "scan" feature is really just taking a photo, so this part of the claim is the same as claim 6. However, this claim goes a step further by asserting clearly

	handwritten text can be searched and edited. No matter how clearly I wrote, the images were treated as photos, and no text in the images was searchable or editable. This would be a very valuable feature if Evernote could make it work reliably.
Post-Conditions	Same as pre-conditions plus one new note containing an image file
Execution History	

Date	Result	Tester
2021-04-25	Fail	James Long

Test Case ID	12
Component	Attachments
Purpose of Test Case	Verify claim that attachments can be exported
Functional Test Type	Specification Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website A note created
Inputs	none
Expected Outputs	Right-clicking to save a file only exports an empty HTML file. This is not a problem because left-clicking the attachment displays a popup menu with the option to save the file. However, there is no way to specify where to save the file (goes to your default download folder) and no opportunity to rename the file (which you can do after you save). If a file by the same name already exists in the default download folder, the file name is automatically appended with a number to prevent overwriting existing files. Since the method of saving is the only discrepancy, and the claim is otherwise supported, I did not consider this a code defect. It might be considered a documentation defect. The article is not app specific. Perhaps having a version of the article for each app version (web, Android, etc.) would be appropriate. I also experimented with the assertion that file names might be meaningless to users depending on how files are attached. After many efforts, I discovered the circumstances to which the statement refers. When a file, such as a photo, is attached in the Android app (and presumably other mobile versions of the app), Evernote sometimes assigns file names. The pattern of when a file is given a name versus when the existing file name is preserved was not immediately apparent. Likewise, the file name given by Evernote can vary greatly in format, and again the pattern is not completely obvious. At first, I thought this was a minor inconvenience. Then I attached two photos in the Android app as a way to transfer them to my PC (something I do often in Evernote). The file names were long and meaningless, and one was valid. But the other was not as it contained a colon. When I tried to save this file in the web client, I got an error. Since there is no way to rename files are part of the save task, I was unable to export the file. However, in the Windows app, I could rename files as I exported them, so a work-around exists. This warrants a defect report.
Post-Conditions	Same as pre-conditions except one or more files will be exported
Execution History	

Date	Result	Tester
2021-04-25	Fail	James Long

Test Case ID	14
Component	Attachments
Purpose of Test Case	Verify claim that text in certain types of attachments can be searched
Functional Test Type	Specification Testing
Pre-Conditions	PC running Windows 10 Brave browser v1.23.71 Chromium 90.0.4430.72 (Official Build) (64-bit) installed Brave shields down for Evernote website A note created with a variety of attachments in common formats for holding text
Inputs	Keywords known to be present in the various documents
Expected Outputs	This test case could be construed as being in the scope of the search feature. However, I am particularly focused on search results for keywords contained within attachments. I expected to see my note in all the search results. However, the only file type that could be searched was PDF. Other files types tested include .docx, .odt, .pptx, .odp, .xlsx, .ods, .json, .rtf, .txt, and .enex. Additionally, the PDF file was only searched if I first selected the notebook containing the note containing the PDF. When searching from Home, no results were found. A defect report is warranted.
Post-Conditions	Same as pre-conditions
Execution History	

Date	Result	Tester
2021-04-25	Fail	James Long

DEFECT REPORTING FORM

DEFECT ID: Evernote.Attachments.1

TITLE: Handwritten Text not Searchable or Editable
REPORTED BY: James Long
REPORTED ON: 04/25/2021

COMPONENT: Evernote
SUBCOMPONENT: Attachment editing
VERSION: Android v8.12.4

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 7

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

When a clearly handwritten note is scanned into the Android app, the text is not searchable or editable.

STEPS TO REPRODUCE:

Handwrite a note clearly
Use the scan feature to capture the note in Evernote

Attempt to search for text in the scanned image
Attempt to edit the scanned image

COMMENTS:

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Attachments.2

TITLE: Cannot Export Due to Invalid File Name
REPORTED BY: James Long
REPORTED ON: 04/25/2021

COMPONENT: Evernote
SUBCOMPONENT: Attachment export
VERSION: web client

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 12

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S3
CONSISTENCY: Rare

DEFECT SUMMARY:

When a photo is attached to a note in the Android app and then saved (exported) in the web client, sometimes the file name prevents export due to invalid characters. The work-around is to use the Windows app to export and rename at the same time.

STEPS TO REPRODUCE:

Attach a photo to a note in the Android app
Access the same note in the web client
Click the photo and attempt to save it

COMMENTS:

Sometimes the file names are valid, but other times they are not. File names generated by Evernote should always be valid for all platforms supported by Evernote.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

DEFECT REPORTING FORM

DEFECT ID: Evernote.Attachments.3

TITLE: Cannot Search Text in Attachments
REPORTED BY: James Long
REPORTED ON: 04/25/2021

COMPONENT: Evernote
SUBCOMPONENT: Attachment search
VERSION: web client

PLATFORM: PC
OPERATING SYSTEM: Windows 10
RELATED TEST CASE IDS: 14

RESOLUTION PRIORITY: P2
RESOLUTION SEVERITY: S1
CONSISTENCY: Regular

DEFECT SUMMARY:

Searching from Home does not return notes containing attachments containing the keywords searched. Searching from a given notebook does not return notes containing non-PDF attachments containing the keywords searched, but results are returned for keywords in PDF files.

STEPS TO REPRODUCE:

Attach .pdf, .docx, .odt, .pptx, .odp, .xlsx, .ods, .json, .rtf, .txt, and .enex files to a note.
Search from Home for keywords that are present in the attachments.
Search from the notebook containing the note containing the attachments for keywords that are present in the attachments.

COMMENTS:

Keywords in the file names return results.

ATTACHMENTS:

1.
2.

For evaluators use only

DEFECT STATUS: _____APPROVED _____ REJECTED

REASON: _____

Automated Testing with Selenium

Automated testing with Selenium can be applied in various ways while using the Evernote web client. I chose to use risk-oriented regression testing to demonstrate how Selenium can be applied. I started by identifying the failures from test cases performed with the web client. Then I filtered out failures that could not be automated with Selenium. For example, one excluded test case was intermittent and difficult to reproduce. Others required multiple devices using non-web clients. The result was a set of six test cases spanning three features. I used the Chrome Dev browser.

Next, I re-ran each test and recorded the steps. During testing, several interesting events occurred. The first occurred upon logging in. I saw a popup indicating I was upgraded to the latest version. When I opened a note, I was given the option to switch back to the Classic editor in Chrome Dev, but Brave did not present that option to me during all of my earlier tests. So, I opened Brave and logged in. The same popup appeared. Likewise, I was able to switch editors. This did not impact my automated testing, but I thought it was interesting as a reminder that systems are perpetually changing. Of course, that has clear implications for testing.

Second, Chrome Dev displayed a popup stating I could search text in PDFs and other documents by upgrading to the Premium license. Again, Brave did not display this message even though popups were allowed. Since one of the failed test cases I planned to automate was "search content in attachment", I had to remove it from my plan. That left a set of five test cases. However, I did not modify my earlier defect report because the development team still needs to know about this defect. Upon repeating my test, the developer (knowing a popup should display) will see that the popup never displays in Brave. That should prompt action to remediate the popup defect.

The third noteworthy event was Evernote denying my attempts to save an article clip or simplified article clip of a complex article page or video article page using Web Clipper due to note size limitations of the Basic license. Chrome's lack of ad suppression resulted in the test pages all exceeding the 25MB limit. I tried to find comparable pages on other news sites, but the same thing kept happening. Consequently, I removed these three test cases from my plan. That left a set of two test cases spanning two features.

After manually modifying the remaining test scripts where needed, the result was a test suite that could quickly validate the fixes to the identified defects. Though my focus was bug regression, test scripts could also be developed for old fix regression and general functional regression. The following table summarizes the features, associated defects, associated test scripts, and expected results of my automated test plan.

Feature	Defect	Description	Test Script	Expected Results
Note Sharing	Evernote.Sharing.1	Cannot See Downstream Account Info	Share1	Email address of second-degree account should be displayed
Attachments	Evernote.Attachments.1	Handwritten Text not Searchable or Editable	Attach1	Keyword should be found and note name should be returned in results

Portfolio_Assignment.side

Structural Testing Analysis

To determine a valid approach to structural testing of the Evernote code, I used the Brave developer tools while logged into the web client to collect some of the code. I was able to see enormous amounts of UI code, but finding the business logic took more effort. Upon inspecting what looked like business logic code, I concluded I would not be able to make heads or tails of it. My coding skills are too nascent for this task. Here is a small sample.

```
/*!
 * ce-123.0.15939 52b13a1c9cf0c33108b8301ccb27dfc44d2f0a8c
 * Copyright 2013-2021 Evernote Corp. All rights reserved.
 * http://www.evernote.com
 *
 */
(self.webpackChunkuno=self.webpackChunkuno||[]).push([[263],{40599:t=>{"use strict";var e=/-
(\w|$)/g,n=function(t,e){return
e.toUpperCase()};t.exports=function(t){return"float"===(t=t.toLowerCase())?"cssFloat":45===t.charCodeAt(0)&&
109===t.charCodeAt(1)&&115===t.charCodeAt(2)&&45===t.charCodeAt(3)?t.substr(1).replace(e,n):t.replace(e,n
)}},15790:(t,e)=>{"use strict";e.E=function(){var t=[],e=t;function
n(){e===t&&(e=t.slice())}return{listen:function(t){if("function"!=typeof t)throw new Error("Expected listener to be
a function.");var r=!0;return n(),e.push(t),function(){if(r){r=!1,n();var
o=e.indexOf(t);e.splice(o,1)}}},emit:function(){for(var
n=t=e,r=0;r<n.length;r++)n[r].apply(n,arguments)}}}},25402:t=>{t.exports=function(t,e,n){var
r,o,i,a,s,l=e.split(".",),u=t;for(r=0;r<l.length;r+=1)if(i=l[r],r===l.length-1&&void
0!==n)if(u[i])for(a=u[i],u[i]=n,s=Object.keys(a),o=0;o<s.length;o+=1)u[i][s[o]]=a[s[o]];else u[i]=n;else
u[i]||(u[i]={}),u=u[i]}},21910:(t,e,n)=>{"use strict";n.d(e,{Z:()=>S});var
r=n(24262),o=n(19013),i=n(13882);function a(t,e){(0,i.Z)(2,arguments);var
n=(0,o.Z)(t),r=(0,o.Z)(e),a=n.getTime()-r.getTime();return a<0?-1:a>0?1:a}function s(t){return
function(t,e){if(null==t)throw new TypeError("assign requires that input parameter not be null or
undefined");for(var n in e=e||{})e.hasOwnProperty(n)&&(t[n]=e[n]);return t}({},t)}var
l={lessThanXSeconds:{one:"less than a second",other:"less than {{count}} seconds"},xSeconds:{one:"1
second",other:"{{count}} seconds"},halfAMinute:"half a minute",lessThanXMinutes:{one:"less than a
minute",other:"less than {{count}} minutes"},xMinutes:{one:"1 minute",other:"{{count}}
minutes"},aboutXHours:{one:"about 1 hour",other:"about {{count}} hours"},xHours:{one:"1
hour",other:"{{count}} hours"},xDays:{one:"1 day",other:"{{count}} days"},aboutXWeeks:{one:"about 1
week",other:"about {{count}} weeks"},xWeeks:{one:"1 week",other:"{{count}}
weeks"},aboutXMonths:{one:"about 1 month",other:"about {{count}} months"},xMonths:{one:"1
month",other:"{{count}} months"},aboutXYears:{one:"about 1 year",other:"about {{count}}
years"},xYears:{one:"1 year",other:"{{count}} years"},overXYears:{one:"over 1 year",other:"over {{count}}
years"},almostXYears:{one:"almost 1 year",other:"almost {{count}} years"}};function u(t){return function(e){var
n=e||{},r=n.width?String(n.width):t.defaultWidth;return t.formats[r]||t.formats[t.defaultWidth]}}var
c={date:u({formats:{full:"EEEE, MMMM do, y",long:"MMMM do, y",medium:"MMM d,
y",short:"MM/dd/yyyy"},defaultWidth:"full"}),time:u({formats:{full:"h:mm:ss a zzzz",long:"h:mm:ss a
z",medium:"h:mm:ss a",short:"h:mm a"},defaultWidth:"full"}),dateTime:u({formats:{full:"{{date}} 'at'
{{time}}",long:"{{date}} 'at' {{time}}",medium:"{{date}}, {{time}}",short:"{{date}},
{{time}}"},defaultWidth:"full"})},f={lastWeek:"'last' eeee 'at' p",yesterday:"'yesterday at' p",today:"'today at'
p",tomorrow:"'tomorrow at' p",nextWeek:"eeee 'at' p",other:"P"};function d(t){return function(e,n){var
r,o=n||{};if("formatting"===(o.context?String(o.context):"standalone")&&t.formattingValues){var
i=t.defaultFormattingWidth||t.defaultWidth,a=o.width?String(o.width):i;r=t.formattingValues[a]||t.formattingValues[
i]}else{var s=t.defaultWidth,l=o.width?String(o.width):t.defaultWidth;r=t.values[l]||t.values[s]}return
r[t.argumentCallback?t.argumentCallback(e):e]}}function h(t){return function(e,n){var
r=String(e),o=n||{},i=o.width,a=i&&t.matchPatterns[i]||t.matchPatterns[t.defaultMatchWidth],s=r.match(a);if(!s)retu
rn null;var l,u=s[0],c=i&&t.parsePatterns[i]||t.parsePatterns[t.defaultParseWidth];return l="[object
Array]"===Object.prototype.toString.call(c)?function(t,e){for(var n=0;n<t.length;n++)if(e(t[n]))return
n}(c,(function(t){return t.test(u)})):function(t,e){for(var n in t)if(t.hasOwnProperty(n)&&e(t[n]))return
n}(c,(function(t){return
```

t.test(u))),l=t.valueCallback?t.valueCallback(l):l,{value:l=o.valueCallback?o.valueCallback(l):l,rest:r.slice(u.length
)}}}var m,p={code:"en-US",formatDistance:function(t,e,n){var r;return n=n||{},r="string"==typeof
l[t]?l[t]:1===e?l[t].one:l[t].other.replace("{{count}}",e),n.addSuffix?n.comparison>0?"in "+r:r+"
ago":r},formatLong:c,formatRelative:function(t,e,n,r){return f[t]},localize:{ordinalNumber:function(t,e){var
n=Number(t),r=n%100;if(r>20||r<10)switch(r%10){case 1:return n+"st";case 2:return n+"nd";case 3:return
n+"rd"}return n+"th"},era:d({values:{narrow:["B","A"],abbreviated:["BC","AD"],wide:["Before Christ","Anno
Domini"]},defaultWidth:"wide"}),quarter:d({values:{narrow:["1","2","3","4"],abbreviated:["Q1","Q2","Q3","Q4"],
wide:["1st quarter","2nd quarter","3rd quarter","4th

In place of actual code analysis, I offer this conceptual analysis. At a high level, the core program accepts a broad range of user inputs through a variety of UI elements (e.g., menus, text editor, search box, etc.), stores them, updates them based on subsequent user input, and outputs them through a variety of UI elements (e.g., text editor, popup windows, search results boxes, etc.). Outputs are primarily screen-based, but printing, email, sharing, and third-party integrations are also valid output ports. Not much computation occurs, but previous inputs are frequently edited. Thus, we might expect the code to be relatively sparse on alternation logic and relatively rich in sequence and iteration logic. For example, the text editor code might prompt for input, store the input, and then repeat for as long as the user is in the text editor. But as we think about what happens within the loop, the number of potential actions is quite high (e.g., store a new attachment, change a font size, change several paragraphs into an unnumbered list, delete a character, etc.). Now we get the picture of a wide, flat loop perhaps entered via a case statement rather than an if statement. This suggests DD-Path testing would be useful.

If we consider the stored data, each note can hold a variety of data elements and can be many MB in size. These are obviously not simple variable declarations in the code. More likely, each note is a complex object such as a document stored in a NoSQL document database. When a new note is created, inputs are stored in the cloud (and also maintained in memory for the user to see and continue manipulating). This represents a definition node in the DU-Path paradigm. Each time the user makes a change, the input is again stored in the cloud. As we already discussed, there may be many paths to accommodate the many types of changes that can be made to a note. Each path would contain a definition node. Each time the user opens a note, switches between notes, or receives a shared note, a usage node is traversed. In a program as data intensive as this, we should expect DU-Path testing to be useful.