

VText: A Plug-in Extension to Add Electronic Textbook Functionality to Microsoft OneNote

John O. Cristy

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Joseph G. Tront, Chair

A. Lynn Abbott

Clifford A. Shaffer

November 15, 2013

Blacksburg, Virginia

Keywords: Tablet, Electronic Textbook,
Active Learning, Microsoft OneNote

VText: A Plug-in Extension to Add Electronic Textbook Functionality to Microsoft OneNote

John O. Cristy

ABSTRACT

Electronic textbooks are different from ebooks (electronic books) in that they allow the user to go beyond just reading material on a computer screen. Electronic textbooks encourage the user to accomplish all of the operations typically performed with a hardcopy text in addition to some functions not possible with paper books. With electronic textbooks users can make annotations in the textbook with e-ink; mark important sections; search over the ink, the text, or even the scanned images; look up items in online dictionaries or encyclopedias; perform interactive reinforcement drills; view simulations; and many other operations afforded by the computing power of the underlying computer and the reach of the Internet connection. These operations encourage students to engage in active reading.

The VText framework is designed to provide many of the desired features of an e-textbook in such a way that it produces pedagogical value rather than just convenience for students. Many so-called e-textbook solutions available today provide few features beyond those possible with hardcopy textbooks. The VText framework is built as an add-in to Microsoft's note-taking program, OneNote. The add-in provides features that facilitate the use of OneNote as a reader and as an educational tool while leaving in place OneNote's strengths in note-taking, collaboration, and search.

Table of Contents

Chapter 1 Introduction.....	1
Chapter 2 Background	4
2.1 Pedagogy	4
2.1.1 Electronic Reading	5
2.1.2 Note Taking	6
2.1.3 Active Reading	7
2.2 Related Work.....	8
2.2.1 Acceptance of Electronic Textbooks.....	8
2.2.2 XLibris.....	9
2.2.3 Kno	10
2.2.4 eReaders.....	10
2.2.5 iPad.....	11
2.2.6 Online Solutions	12
Chapter 3 VText Features.....	14
3.1 OneNote	15
3.2 Book View.....	18
3.3 Navigation	19
3.4 Searching the Web	22
3.5 Notes Page.....	23

3.6 Classroom Sync.....	24
3.7 Bluetooth Mobile Scanner.....	25
3.8 Content Importers.....	27
3.8.1 Portable Document Format to OneNote (PDFON)	27
3.8.2 Rich Text Format to OneNote (RTFON)	28
3.8.3 PDF Rasterized to OneNote (PDFRON).....	28
3.8.4 PDF to Watermarked OneNote (PDFWON).....	29
3.8.5 Text to OneNote (TXTON)	29
3.9 Distribution of Content.....	29
3.10 Extensibility and the eQuiz System	30
Chapter 4 Implementation.....	33
4.1 Technologies	33
4.1.1 .NET	33
4.1.2 C#.....	34
4.1.3 Java	34
4.1.4 XML	35
4.1.5 Windows Presentation Foundation.....	35
4.1.6 OneNote API	35
4.1.7 Android.....	36
4.1.8 Visual Studio	37
4.2 Libraries	37
4.2.1 Win32	38
4.2.2 Ghostscript and GhostscriptSharp	38
4.2.3 XDesigner RTF.....	38
4.2.4 iText.....	39

4.2.5 DotNetZip.....	39
4.2.6 32feet.NET	39
4.3 VText Components	40
4.3.1 Initialization & Shutdown	40
4.3.2 System Thread	41
4.3.3 Ribbon	42
4.3.4 Window Manipulation.....	43
4.3.5 Page Cache	45
4.3.6 Navigation	46
4.3.7 Gestures	49
4.3.8 Settings	50
4.3.9 Right Click Menu	51
4.3.10 Web Search.....	52
4.3.11 Notes Page	52
4.3.12 Bluetooth Mobile Scanner	52
4.3.13 Classroom Sync	55
4.3.14 RTF Vector Graphics Rasterization	57
4.3.15 RTF Importer (RTFON).....	57
4.3.16 PDF Importer (PDFON).....	57
4.3.17 PDF Rasterized to OneNote (PDFRON) and PDF to Watermarked OneNote (PDFWON) Importers	58
4.3.18 Text Importer.....	59
4.3.19 eQuiz System.....	59
4.4 Implementation Summary	60
Chapter 5 Performance	61

5.1 Page Cache Lookup.....	62
5.2 Generating the Page Cache	62
5.3 System Thread.....	63
5.4 Power Usage of VText.....	63
5.5 Bluetooth Scanner Performance.....	64
5.6 Performance Summary.....	65
Chapter 6 Usability Study and User Assessment.....	67
6.1 Pre-Survey.....	67
6.2 Post-Survey	72
6.3 Assessment Conclusions	76
Chapter 7 Conclusions.....	77
References.....	78
Appendix A User Guide	82
A.1 VText.....	82
A.2 System Requirements	82
A.3 Installation/Uninstallation/Upgrade	83
A.3.a Install.....	83
A.3.b Uninstall.....	85
A.3.c Upgrade.....	86
A.4 VText Controls.....	86
A.4.a Overview of New Tab “VText”	86
A.4.b Overview of Settings.....	87
A.5 Controlling the Windows	88
A.6 Basic Navigation	88
A.7 Pins Page	90

A.8 Search the Web.....	91
A.9 Put In Notes	92
A.10 Classroom Sync.....	92
A.11 Bluetooth Scanner	93
A.12 Importing Content	95
A.12.a PDF to OneNote (Preserving text as text).....	95
A.12.b Import PDF (One image per page)	97
A.12.c PDF with Watermark Importer	97
A.12.d RTF to OneNote	99
A.12.e TXT Importer.....	100
A.13 Troubleshooting.....	101
A.14 Other Resources	102
A.15 Release Notes	102
Appendix B Pre-Survey Results	104
Appendix C Post-Survey Results.....	115

List of Figures

Figure 3.1 OneNote's Ribbon with VText additions (Used with permission from Microsoft.)	16
Figure 3.2 The OneNote and VText Right Click Menus (Used with permission from Microsoft.)	16
Figure 3.3 OneNote Navigation (Used with permission from Microsoft.).....	18
Figure 3.4 Side by Side View (Used with permission from Microsoft.)	19
Figure 3.5 VText's Gestures (Used with permission from Microsoft.)	20
Figure 3.6 Navigation Buttons (Used with permission from Microsoft.).....	21
Figure 3.7 The navigation buttons sliding menu with book content in the background (Used with permission from Microsoft.).....	22
Figure 3.8 The Pinned Pages View (Photo by author 2013).....	22
Figure 3.9 VText's Search the web feature (Used with permission from Microsoft.).....	23
Figure 3.10 A notes page with clippings and links back to the source pages in the textbook (Used with permission from Microsoft.)	24
Figure 3.11 Classroom Sync.....	25
Figure 3.12 The Bluetooth Mobile Scanner including the screen seen on a typical Android phone on the left with an application window containing a captured and transferred image on the right. (Photo by author, 2013).....	25
Figure 3.13 Bluetooth Scanner Android Application (Photo by author, 2013).....	26
Figure 3.14 Example of PDF Importer with user selected text.....	28
Figure 3.15 An Example of an embedded C++ File in OneNote (Used with permission from Microsoft.)	31
Figure 3.16 Quiz file included on a OneNote Page (Used with permission from Microsoft.)	31
Figure 3.17 The Quiz program (Photo by author, 2013)	32

Figure 4.1 VText's Initialization and Shutdown Process	41
Figure 4.2 Diagram of Window Layout including Z-Order	45
Figure 4.3 The VText Page Cache	46
Figure 4.4 Navigation Buttons Automatically Adjusting Position	47
Figure 4.5 Bluetooth Scanner Communications Outline	53
Figure 4.6 Bluetooth Scanner Android Process	55
Figure 4.7 Multicast used to send page synchronization beacon.....	56
Figure 5.1 Power Usage of VText	64
Figure 6.1 Student Preferences Towards Paper and Electronic Reading.....	71
Figure A.1 Installation Wizard (Used with permission from Microsoft.)	83
Figure A.2 Installation Options (Used with permission from Microsoft.)	84
Figure A.3 Index pages progress window (Photo by author, 2013)	84
Figure A.4 OneNote Menu (Used with permission from Microsoft.)	84
Figure A.5 OneNote Menu (Used with permission from Microsoft.)	85
Figure A.6 OneNote Add-Ins menu (Used with permission from Microsoft.).....	85
Figure A.7 Uninstallation Procedure (Used with permission from Microsoft.)	86
Figure A.8 Ribbon tab added to OneNote by VText (Used with permission from Microsoft.)	86
Figure A.9 VText Gestures (Used with permission from Microsoft.).....	89
Figure A.10 Navigation Buttons (Used with permission from Microsoft.).....	89
Figure A.11 Navigation Buttons Menu (Photo by author, 2013)	90
Figure A.12 Methods of opening the pinned pages window (Photo by author, 2013).....	91
Figure A.13 The pinned pages window (Photo by author, 2013).....	91
Figure B.14 Search the web from the right click menu (Used with permission from Microsoft.)	92
Figure A.15 Copy text to notes from the right click menu (Used with permission from Microsoft.)	92
Figure A.16 Elements of the Classroom Sync feature (Photo by author, 2013).....	93
Figure A.17 Get picture from Bluetooth button (Photo by author, 2013)	94
Figure A.18 Android application launcher (Photo by author, 2013)	94
Figure A.19 Bluetooth Scanner device select (Photo by author, 2013).....	94

Figure A.20 Bluetooth Scanner take picture (Photo by author, 2013).....	94
Figure A.21 Bluetooth Scanner image editing tools (Photo by author, 2013).....	95
Figure A.22 Bluetooth Scanner interface with cropped photo (Photo by author, 2013)	95
Figure A.23 Method for opening the Import Wizard (Photo by author, 2013)	96
Figure A.24 Import Wizard (Photo by author, 2013)	96
Figure A.25 Results of selecting add page numbers (Used with permission from Microsoft.)	96
Figure A.26 Example of the Import Wizard Progress Bar (Photo by author, 2013)	97
Figure A.27 Import Wizard (Photo by author, 2013)	97
Figure A.28 PDF to VTI Tool (Photo by author, 2013)	98
Figure A.29 The PDF to Watermarked VTI	98
Figure A.30 Import Wizard (Photo by author, 2013)	99
Figure A.31 VText Word Ribbon (Photo by author, 2013).....	99
Figure A.32 VText Import Wizard (Photo by author, 2013)	100
Figure A.33 VText Import Wizard (Photo by author, 2013)	101
Figure A.34 VText version information (Photo by author, 2013)	103

List of Tables

Table 1 Performance of Bluetooth Scanner	64
Table 2 Data Relating to Student Collaboration	68
Table 3 Student Opinion on Impact of Technology on Education	69
Table 4 Reasons for Using Electronic Textbooks.....	70
Table 5 Reasons for Not Using Electronic Textbooks.....	71
Table 6 Student Satisfaction with VText Features	73
Table 7 Importance of Features in VText	74
Table 8 Cross-tabulation of VText Use and Interest in Using Electronic Textbooks in the Future	75

Chapter 1

Introduction

Present day textbooks have evolved greatly from their black and white, mostly text predecessors. The first step in the evolution of textbooks to electronic form has been to convert paper pages to page images on a screen. While publishers have subsequently added supplemental content which uses improving technology, such as increased storage, greater processing power, and the Internet, the hardcopy books have lagged behind in capitalizing on improving technology. Predictions have been made since the 1980s that sales and use of computerized solutions would surpass those of printed textbooks in just a few decades [1]. However, while electronic book sales have seen rapid growth, electronic textbooks sales have grown far slower [2, 3].

There is evidence that the reason electronic textbook adoption has been slower than predicted is due to poorly developed products [1, 4, 5]. As textbooks transition to electronic form, they should maximize the potential for improved learning, not just potential corporate profits [1, 6]. To do so, factors known to improve learning outcomes must be considered. These factors include providing students with unconstrained note taking and the ability to find related information to allow active reading, while not discouraging the use of such a tool by presenting the material in a difficult to use package.

Many efforts have been made to produce an electronic textbook platform in the past two decades; however, they have been subdued by uninterested students and faculty, technological difficulties, and economic complexities. Low interest can be explained by the tendency of students to not change study habits [7]. Problems due to hardware, such as low resolution

screens, poor battery life, and insufficient computing resources, have caused reduced user satisfaction with the devices. Fortunately, most technological issues have been solved with the advent of more modern hardware. However, these solutions are also considered unfavorable when students must purchase new hardware or when the electronic book's prices, terms, and conditions do not compare favorably with the printed edition.

This thesis describes the features and design of an electronic textbook framework add-in for Microsoft OneNote known as VText. We call the system a framework since VText does not provide the content of the electronic textbook but rather the facilities that allow the user to appropriately interact with content to better understand and recall the information provided by information authors. To guide choosing targeted features for electronic textbooks, a discussion at the 2010 Workshop on the Impact of Pen Technology in Education [8] that identified desirable electronic textbook features was considered along with lessons learned from previous electronic textbook designs. The following lists some of the identified desirable electronic textbook capabilities:

- Electronic notes with electronic ink/highlighter; stylus enabled
- Audio/video playback
- Internal search including on electronic ink
- External internet search
- Customizable content from publishers
- Customizable content by user
- Embedded simulations
- Hyperlinks to the Internet
- Self-updating content
- Drill down exercises
- Bookmarks
- Text to speech
- Collaboration
- Split screen capability
- Voice recognition control
- Individual tutoring

VText, through OneNote and components of the add-in, provides and facilitates the inclusion of many of the features listed above in electronic textbooks. VText allows authors to suggest and students to easily find documents, including multimedia, embedded in and beyond the textbook through hyperlinks and Internet search. VText's implementation allows the textbook to interact with any program that can run on a Windows-based machine, allowing external programs to provide further and more specialized content. VText also does not limit the type or style of notes a student may wish to take, be they with electronic ink or with typed text. VText addresses issues identified from previous efforts [9, 10, 11] by providing a mechanism to display content and transfer knowledge in a form familiar to experienced students while providing abilities known to improve learning outcomes.

This thesis is organized into seven chapters including this introduction. Chapter 2 discusses the results of previous research and successes and failures of earlier solutions. In Chapter 3, the features and user operation of VText are explained. Chapter 4 describes the implementation details of VText. Chapter 5 analyzes the performance characteristics of the add-in and design decisions. Chapter 6 discusses the results of surveying users of VText. Finally, Chapter 7 summarizes the results of this work and provides suggestions for future efforts to improve this tool.

Chapter 2

Background

As more and more information and communication is available through computer networks, and information repositories, educators should prepare students for effective electronic reading [12, 13]. Further, as students year after year are more comfortable with electronics, pedagogies which were optimal in the past must be adapted, while pedagogies which were not well accepted in the past may be worth re-exploration. For these reasons, but also financial gain, book sellers and publishers have attempted to create electronic textbook devices. These readers include both hardware and software solutions. This chapter consists of two sections. The first presents research relevant to the pedagogical design of electronic textbooks, while the second section discusses solutions currently or previously in development or on the market.

2.1 Pedagogy

While electronic textbooks offer many physical benefits, such as reducing publishing and distribution costs as well as avoiding carrying many pounds of books, they also provide potential to improve study outcomes through improved pedagogical practices. Research shows that students must be able to focus and contemplate the material they are reading. The latter is known as active reading. Electronic textbooks may improve students' efficiency, reducing the time of or increasing the content covered in study sessions. There is some indication that students are more likely to complete readings [14] and possibly in less time when using electronic textbooks instead of traditional textbooks, although other factors may have influenced these findings [15]. Another critical component of active reading involves being able to take notes which are derived from complete and complex thoughts related to the material. In order to

achieve active reading, the student must be able to focus on considering the content, rather than overcoming obstacles produced by the presentation and delivery of the material.

2.1.1 Electronic Reading

Electronic textbook designers must concern themselves with the ability of a user to comfortably and effectively read from a computer screen. Work by Aust [16], Rockinson-Szapkiw [11], Daniel [3], and Shepperd [15] indicate that university students reading from computer screens performed academically as well as those who used traditional paper textbooks. It is important to note that in the studies of Rockinson-Szapkiw, Daniel, and Shepperd the electronic textbooks were merely electronic versions of the paper textbooks with few extra features. These works all showed that students who studied with an electronic version of the material performed as well as those with a paper version. However, there does seem to be cause for concern as students who used the electronic version in Shepperd's study disliked the experience, and further study by Daniel found students were frequently distracted by other activities on their computer, such as games, e-mail, and Facebook.

Interestingly, Shepperd's electronic users spent less time reading, while Daniel's electronic users spent more time reading. Daniel suggests students take longer as they are distracted, while Shepperd suggests students spent less time reading since they did not like reading from the electronic reader. However, this explanation for a decrease in time spent studying would suggest that the readers of the traditional book would have performed better in the class. Changes in student computer behaviors during the four years between Shepperd's study in 2008 and Daniel's study in 2012 may also explain the differing outcomes. Further confounding any conclusion on differences in reading efficiency, Rockinson-Szapkiw's work found there was no difference in time spent by either group. The study did find that students who chose electronic textbooks self-assessed a higher level of learning and reported a higher satisfaction level with the course. While these results are neither encouraging nor discouraging, these studies leave open opportunities for testing with more advanced textbook platforms and with students more inclined to choose technology.

While these tests all dealt with university level students, work by Maynard investigated the impact of electronic textbook use on educating 11-12 year old students [14]. There is substantial

research indicating that older students are unlikely to change their study habits regardless of potential benefits [7]. This suggests that electronic textbook adoption may be more likely to occur with students who are more used to computers being a ubiquitous part of their education. Maynard's study utilized an electronic textbook that included a glossary, hypertext, and multimedia. Students who used the electronic textbook outperformed the other students on individual and group testing. In contrast to Daniel's study [3], Maynard [14] found that the students preferred using the electronic textbook and like in Rockinson-Szapkiw's study [11], the students showed increased desire to learn.

While electronic textbook studies have yielded few conclusive results in university settings, anecdotal evidence indicates that younger students seem to be more accepting. This may indicate that there is a generational shift or that experienced students are unwilling to change their habits. However, these studies do indicate that electronic textbooks are not detrimental and that well designed solutions may have pedagogical potential for future students.

2.1.2 Note Taking

Note taking is a key component in active reading and learning. Note taking is the first step towards processing information, and is not simply a way to store information on paper [17]. It is also known that a greater quantity of notes that represent complete thoughts correlates with increased learning. The process of taking complete notes increases the time that ideas spend in working memory being subconsciously or consciously analyzed and refined. This allows for better understanding and deep storage. Given the importance of this process, this idea of complete and complex note taking should be supported by digital note taking systems [18].

Digital systems should not add extra burden to students when taking notes. Input systems with difficult learning curves, such as Palm OS's Graffiti [19] commonly found on personal digital assistants (PDAs), should be avoided [18]. While typing can be faster for text, systems should include stylus input to allow for free-form charts and diagrams. Further, even with free-form notes, small screens and difficult navigation should be avoided. These extra obstacles reduce the number of notes written as well as the complexity and completeness of notes written in comparison to students who use paper and pen. Tablet PCs can be designed to reduce the small screen and input obstacles by providing screens approximating paper in size and stylus

input. Kim's results indicate that although tablet PCs removed these obstacles, there was still a learning curve and the screen was still too small for some students. Students who found the screen too small tended to write larger than they would with a normal pen; however, this may have been due to the precision of the stylus technology used. Kim concludes that a system that allows for both typed and hand written notes is preferred by students [18].

Note taking is most effective when students utilize complete thoughts. Because note taking, including free-form notes, is critical for students to perform active reading, it should be facilitated by an electronic textbook. Electronic textbook designs should provide these capabilities in such a way that they do not limit note taking by the student.

2.1.3 Active Reading

Active reading involves performing critical thinking while reading. Reading in this manner is an integral part of studying [20]. This method for learning can be facilitated by making notes in the book, but more importantly by imagining notes as a conversation with the author [12, 21]. Such activity allows for easier recall, but also improves understanding as it forces the reader to slow down to think. Schilt discusses the importance of annotations as a method of taking notes while reading. Annotations provide a connection to ideas that were formed while reading the annotated content. However, at times, separate notes are important as annotations can be lost within the larger document. Thus these two types of markup information should be available to the electronic textbook user.

Another key to active reading is being able to find related materials. This area of active reading involves looking deeper into topics often by reading other documents. Finding a suggested external document can be time consuming. However, with computers and the Internet, this task has been greatly sped up [20]. Streamlining the process of finding new information through hyperlinks increases the number of references actually viewed by students [16, 22]. While authors may suggest further reading, the reader's interests and understanding can change which documents should be suggested. Computers can intelligently suggest relevant materials from within and outside of the current document [16, 12]. In research performed by Aust, even though students read more documents, there was a reduction in total study time due to the

increased speed of finding external documents. This ability to explore topics more deeply is particularly important in education [22].

Adler et al. looked into the design of active reading devices to support different occupations [23]. A key problem in design is that different subject matters and goals change the requirements of a device. Much like the results of the previously mentioned work, they found that many readers were writing and reading in different documents at the same time. This use pattern suggests that multiple-document displays and hyperlinks are both beneficial. While Adler's work looked beyond education, many of the same principles apply particularly when addressing the needs of students studying different subjects or performing different tasks, such as studying or completing assignments.

2.2 Related Work

Interest in developing electronic textbooks has increased as use and ownership of computers has spread. Publishers have created simple solutions while other companies have attempted to make electronic textbook readers which utilize the potential of computers. However, no solution has achieved widespread deployment.

2.2.1 Acceptance of Electronic Textbooks

Electronic textbook sales have not gained a large portion of the market, making up just three percent of total textbook sales as of 2010 according to the National Association of College Stores [24]. Research suggests that university students choose paper books because they are familiar and allow them to highlight text, dog ear pages, and take notes [11, 24]. Some students are also concerned that hardware and software may stop working. Further, current options do not provide the value students are looking for in their learning support materials. Student Public Interest Research Groups technical director Nicole Allen claims that because "Students grew up learning from print books ... it's not surprising that they carry a preference for a format that they are most accustomed to" [24]. On the same topic Tracey Weber, Barnes and Noble's executive vice president for textbooks and digital education, says "The real hurdle is getting them to try it" [24]. These viewpoints align with earlier work, which showed that students are resistant to changing their study habits [7]. However, research shows that younger university students are more

inclined to try the devices [2], which suggests the change may be coming, but will be more gradual as indicated by another study by Maynard [14]. However, on the negative growth side, electronic textbook use may be discouraged by the currently available options as well as poor perception among the current generation of university students. The future of the electronic textbook movement is at an important juncture in its potential growth curve.

2.2.2 XLibris

XLibris is an early example of an active reading machine first presented in 1998 [20, 25]. This device alleviated many concerns with electronic reading, including tangibility, free-form ink notation, page orientation, access to multiple documents at once, sharing, and legibility of computer screens. XLibris attempted to create a document reader which could be used like paper with free-form note taking. Great care was afforded to the user interface to reduce the learning curve. The device employed physical buttons to perform page changes while using a minimalistic interface to reserve screen space for the content. The interface did provide some indication to the user to indicate where in the document they were currently reading. Navigation could be performed linearly with physical buttons, thumbnail view, and a bookmark-annotation hybrid. To further support active reading, XLibris attempted to provide related documents the user would likely be interested in by using keywords. To perform this estimation, XLibris used the content of the reader's own annotations on the page. It then provided links to other documents in the margin that are related to the original document and the keywords from the annotations that can be viewed at the reader's convenience. This allows the reader to discover new content seamlessly while not requiring the user to immediately switch to the suggested document.

XLibris was designed to run on a 12.1" tablet tethered to a Windows 95/NT desktop computer. XLibris would shrink documents to fit on the tablet screen. The XLibris spawned a series of further research into analyzing annotations in meaningful ways for suggesting external documents both locally and online based on what the user had written. Ultimately, the project was not continued.

2.2.3 Kno

The Kno tablet was a promising full-featured e-textbook hardware and software solution [26, 27]. There were originally two systems proposed, one with a 14.1 inch touch screen and the other with two of the screens hinged together. This form factor would have allowed a paper book-like experience that included all of the benefits of a computer. The hardware also allowed for Bluetooth devices such as a keyboard and mouse to connect to the system. The software for textbooks included many features such as note-taking, highlighting, and search that would have made the Kno fit many of the requirements of a fully featured tablet electronic textbooks. There was even a planned SDK for developers to extend the system including Google Documents (now Drive) and Microsoft Office Live support. Unfortunately, the project was cancelled in April 2011 in favor of a software-only approach [28]. The current Kno software is web-based and includes annotations and highlighting, but no pen support. In this implementation, there is also no easy, built-in way to view two pages at one time, such as the two-screened version of the tablet would have allowed.

2.2.4 eReaders

While eReaders, such as the Amazon Kindle, Sony Reader, and Barnes & Noble Nook, have gained popularity for reading novels, their use with textbooks has not, although several attempts have been made to fit these devices into this application arena. eReaders typically use electronic paper that attempts to look like normal paper with high contrast, backlight free displays. Colleges, publishers, and eReader manufactures have shown interest in deploying these devices for textbooks. Seemingly, reducing the cost of textbooks has been a major motivator in these efforts [5, 4].

A pilot study at Northwest Missouri State University tested the Sony Reader for textbook display. Unfortunately, the students quickly asked for paper textbooks shortly after the switch. Students had complained primarily that the device made navigating in the textbooks difficult [4]. One year later in 2008, a pilot study was performed to test the Kindle DX, a 9.7" tablet using E Ink's electronic paper similar to the Sony Reader, at many schools across the country. A study at Reed College found that the Kindle DX provided a usable form factor, legible display, long battery life, convenient delivery mechanism, and paper savings [5]. However, there were few

pedagogical improvements. The Kindle DX suffered flaws similar to the Sony Reader. These flaws include missing PDF capabilities such as annotation and highlighting, difficult distribution of reading materials beyond the course text, poor image display due to low resolution and lack of color, slow changing of pages and books, difficulties for the visually impaired, and cumbersome note taking. Students noted that they were not likely to purchase the device if given the option unless it were less than \$100, far less than the Kindle DX's \$489 price at the time of the study. However, students indicated they would be interested in a system that ran on or replaced devices they already owned, specifically laptops. In both cases, as well as similar pilot studies, students and faculty overwhelmingly rejected the devices, but were optimistic for future possibilities [4, 5, 10].

2.2.5 iPad

Apple's iBooks textbooks for the iPad is an improvement upon the Kindle, Sony Reader, and Nook as it provides full color on a device with greater computational power. Further, as suggested in the Reed College and Northwest Missouri State University study results, the iPad's cost may be more likely to be accepted by students as the iPad offers more functionality than the eReaders. iPad textbooks offer to publishers and authors the capabilities to include interactive images, three dimensional images, and chapter review quizzes. Apple offers publishers an authoring tool to create these multimedia textbooks. Textbooks are distributed through iTunes U, an extension of the popular iTunes content service.

Reed College conducted a pilot study using the iPad for course readings in a senior level political science course [9]. The results were generally more positive than the Kindle study's. They found that the LCD touch screen did not cause discomfort for readers, while touch was far more responsive than the controls had been on the Kindle DX. They also found that while there were some negative comparisons to the Kindle DX, such as shorter battery life and greater weight, these differences were marginal. Students in the study reported they would be more likely to forgive the cost of the iPad or a similar device since it was not a single function device, although they considered the potential for distraction to be a disadvantage. Another key finding was the preference for the ability to take free form notations which was not possible with the Kindle DX, even though no students reported using a stylus. Similar to the Kindle DX, they found the keyboard to be inadequate for academic work. In contrast to the results of other

studies, students reported that they hoped to use the iPad again for academic work and would recommend the device to friends.

2.2.6 Online Solutions

Many publishers have developed online systems for displaying textbooks. These often come with the requirement that a student be connected to the Internet, although some support a check out feature which allows the book to be temporarily locked to one device but does not require that device to be online. Some of the more popular products are CourseSmart [29], VitalSource [30], Chegg [31], Flatworld Knowledge [32], and Kno [27].

CourseSmart is an electronic textbook provider founded by Pearson, Cengage Learning, McGraw-Hill Education, Bedford, Macmillan, and John Wiley and Sons. CourseSmart textbooks can be read in most web browsers and can be checked out for offline reading. The textbooks allow some annotation, but not free form ink notations. While the catalog includes many popular textbooks, the textbooks are rental only. Most of the other online solutions are similar.

VitalSource's Bookshelf allows users to read textbooks on desktops, laptops, and mobile devices. Northwest Missouri State University tried VitalSource's Bookshelf after the Sony Reader test (see Section 2.2.4). Students can annotate and highlight text, but not free-form with a stylus. This is much like CourseSmart's annotation capability. At the end of the first semester, a survey indicated that 40% of students studied less because VitalSource made studying more difficult [4]. However, students did point to some positive features. For instance, they found that searching and being able to copy text to other programs was beneficial. Overall, Bookshelf has not achieved widespread use.

Flatworld Knowledge is notable as its books are provided free and open source. This allows faculty to create their own versions of textbooks as it best suits their course. In order to take notes, highlight, and view study materials, a student must purchase a \$19.95 account. There have been few studies to measure the pedagogical capabilities of these open source textbooks.

Generally, it seems these online solutions do not meet the requirements of students. They are missing features like free form note taking and offer little more in terms of content, with some

exceptions, than PDFs and CD-ROM companions for paper-based books. While there is plenty of content available for these products, it is not presented in a useful way.

Chapter 3

VText Features

As one possible solution for providing benefits to students beyond that of traditional textbooks, VText should equip students with the ability to easily read material, annotate content, find content quickly and easily, view multimedia, expand the material via external enrichment content, collaborate with other students, and provide access to course materials during class lectures. While many great features can be included, content must also be available. This requires that publishers and authors be able to create new content as well as convert older content into the format usable on this new platform.

VText is designed to meet the requirements of both publishers and students. While some requirements are inherently satisfied by OneNote, others are fulfilled by VText operating as an add-in to OneNote. By utilizing VText, students can read textbook content on their computers and mobile devices while creating and sharing annotations with their classmates. Reading on the computer screen is enhanced by the availability of gestures included in VText which allow for bookmarking and smoothly transitioning between pages. Deeper exploration and explanation is improved by allowing students to find further explanation of key concepts via VText's interface with Google, Bing, and other search engines. Studying for assessments can be facilitated by VText which allows students to create a summary page for books or chapters. During lectures, students are able to follow along with their instructor by using the Classroom Sync feature of VText. Students can also quickly incorporate charts and diagrams, even those drawn on a whiteboard or chalkboard, in their notes accompanying the textbook via the Bluetooth Scanner feature. In order to encourage publishers to provide content, VText provides mechanisms for

importing other digital formats (other than OneNote .one files), such as Portable Document Format (PDF) and Rich Text Format (RTF), to OneNote with minimal effort.

This chapter lists and describes in detail the features of VText. The pedagogical purpose of each feature is included as well as the intended user experience.

3.1 OneNote

To hasten development of the electronic textbook framework, Microsoft's OneNote was used as the basis for VText. OneNote 2007 and later versions provide a complete extensibility framework to allow for add-ins (OneNote 2003 offered a more limited Application Programming Interface, or API). APIs allow programs to interact with each other in a well-defined manner such that the program offering the API does not need to know exactly what other programs intend to do. Add-ins can be used to control both page navigation and content. By using general Window's APIs, windowing, external programs, and gestures can be manipulated and accessed programmatically to increase the potential features of the add-in. This section primarily describes native OneNote behaviors that are relevant to an electronic textbook and begins to relate how these features are enhanced with the addition of the VText add-in.

OneNote organizes content into a multi-section book-like hierarchy. The intention is to provide notebooks containing sections and subsections, which in turn consist of pages. VText uses this hierarchy to provide books and chapters. Both sections and pages can be dynamically increased in size to allow for additional content as the user reads and works their way through content. For instance, a student may add an extra page to an existing chapter of a textbook in order to include related notes from other sources such as lectures, while an instructor may add another section from another book or a case study description to distribute to students. This flexibility allows instructors to better merge lectures and textbook content in order to simplify the students' studying efforts by providing a single destination to house many of the resources needed for a specific subject.

OneNote 2007 and later versions utilize Microsoft's Ribbon interface. The ribbon organizes controls into a hierarchy of tabs and groups. VText adds controls to the Ribbon in order to provide access and control to its many features. The main VText tab includes four groups of

controls. The View group adjusts how the windows are laid out and if navigation buttons should be presented. The Controls group provides access to actions the add-in can perform. The Options group allows users to configure how the add-in performs. Finally, the Classroom Sync group allows access to controls for this special synchronizing feature. These additions are illustrated in Figure 3.1.

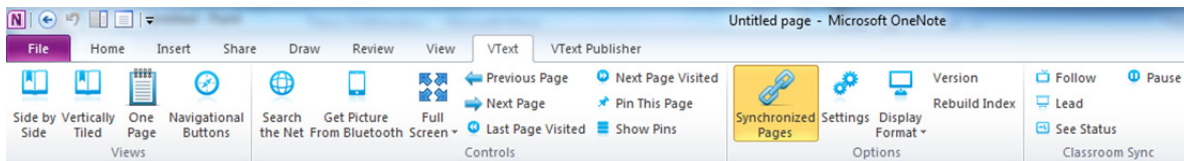


Figure 3.1 OneNote's Ribbon with VText additions (Used with permission from Microsoft [37])

OneNote, like many programs, uses a pop-up context menu when the user right clicks on an item or portion of a page. This allows the user to perform actions related to the content selected or near the click. VText adds a separate menu to offer its context based options. These include searching the web and placing the selected text on a notes and summary page. This additional menu is shown in Figure 3.2 . The buttons are shaped for easy use with touch enabled devices.

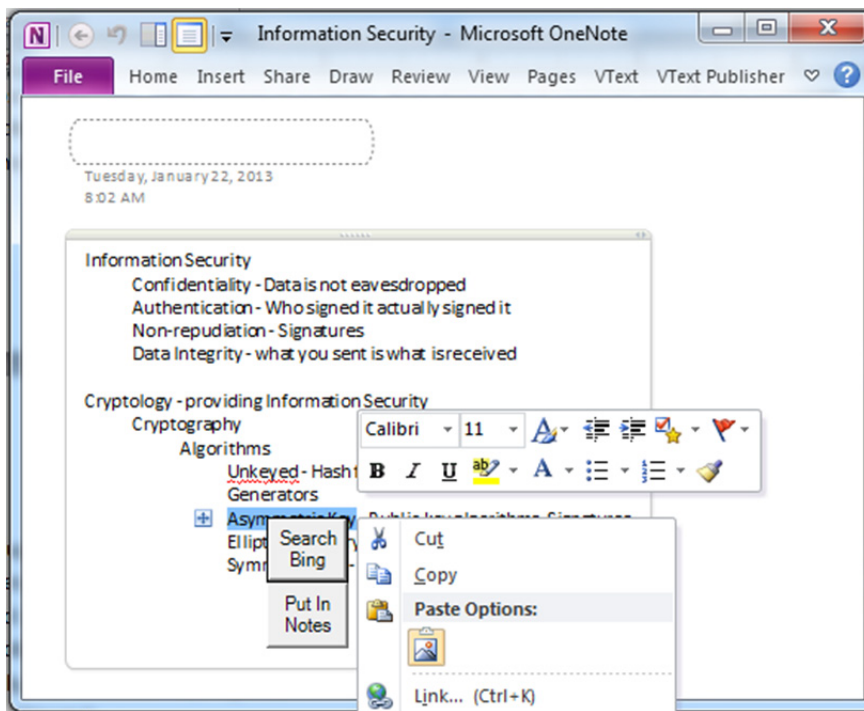


Figure 3.2 The OneNote and VText Right Click Menus (Used with permission from Microsoft [37])

OneNote allows for a variety of page content. Content can include traditional digital text, images, digital ink, and files. Files of any type can be embedded into the page and opened via the appropriate program. This can include, but is not limited to, audio and video files. VText provides access to an accompanying quiz program and could be used with a Windows program. OneNote enhances images and digital ink, by allowing users to search for words included in scanned documents and inked notes. OneNote's search quickly finds any text in any notebook. Audio and video can be recorded and saved to a page through OneNote's interface. OneNote's search includes the ability to index and search audio content. This is useful for students who wish to record lectures for later review.

OneNote also allows for collaboration via shared notebooks. This allows many users to make changes to the same notebook. An instructor may decide to use a shared notebook for a course's textbook to allow students to work together in generating notes, collecting external sources, and planning for course projects. VText's Classroom Sync feature can be used during project meetings, study groups, and lectures to ensure everyone is on the same page.

Navigation through a book in native OneNote consists of selecting pages from a list. While this method provides access to every page at once, the list of pages and the list of notebooks and sections take up a large portion of the screen. This view is shown in Figure 3.3. OneNote includes a full screen view that removes these lists, as well as minimizing the ribbon, to increase screen space available for displaying content. OneNote also includes touch gestures allowing the user to pan and zoom on open pages. Saving screen space is particularly important when a user wishes to have two pages open at the same time. The process of switching to a dual paged view is simplified by VText.

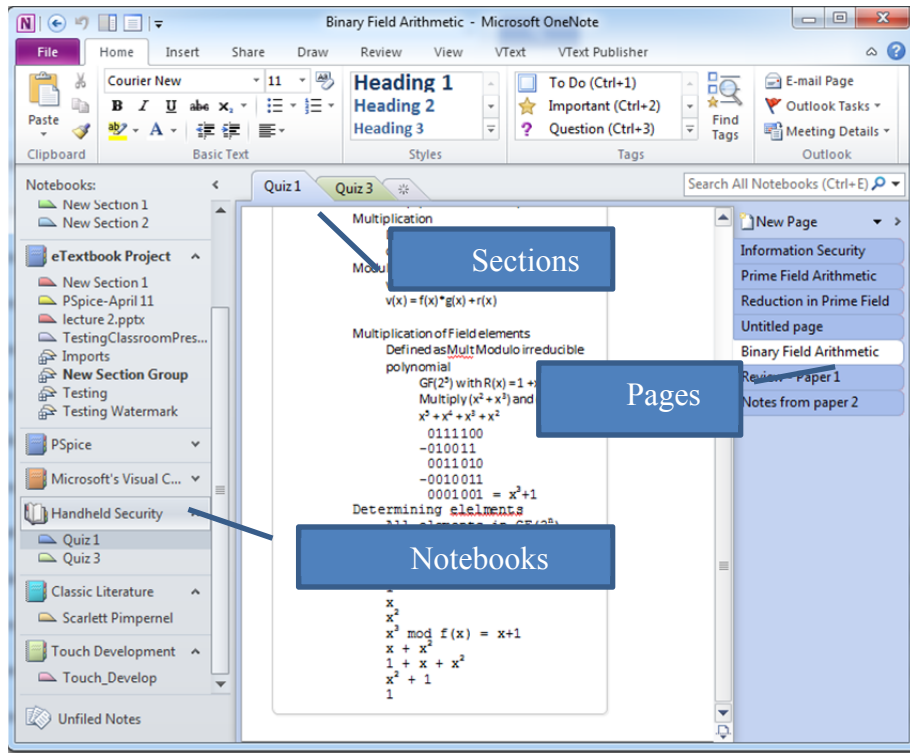


Figure 3.3 OneNote Navigation (Used with permission from Microsoft [37])

3.2 Book View

In order to provide an option that offers a style of reading more familiar to most students, VText includes the ability to automatically open consecutive pages in a book in two separate windows side-by-side or top to bottom. The two windows can also be desynchronized to allow non-consecutive pages or pages from different books to be viewed simultaneously on the screen. This view also automatically changes OneNote to full screen view which reduces the amount of screen space reserved for OneNote's functions in order to display more content. The result of this function is shown below in Figure 3.4. While this mode is currently limited to showing two pages on the screen, it could be readily modified to display more than just two pages to emulate the effect of having multiple books open to various pages on a physical desktop.

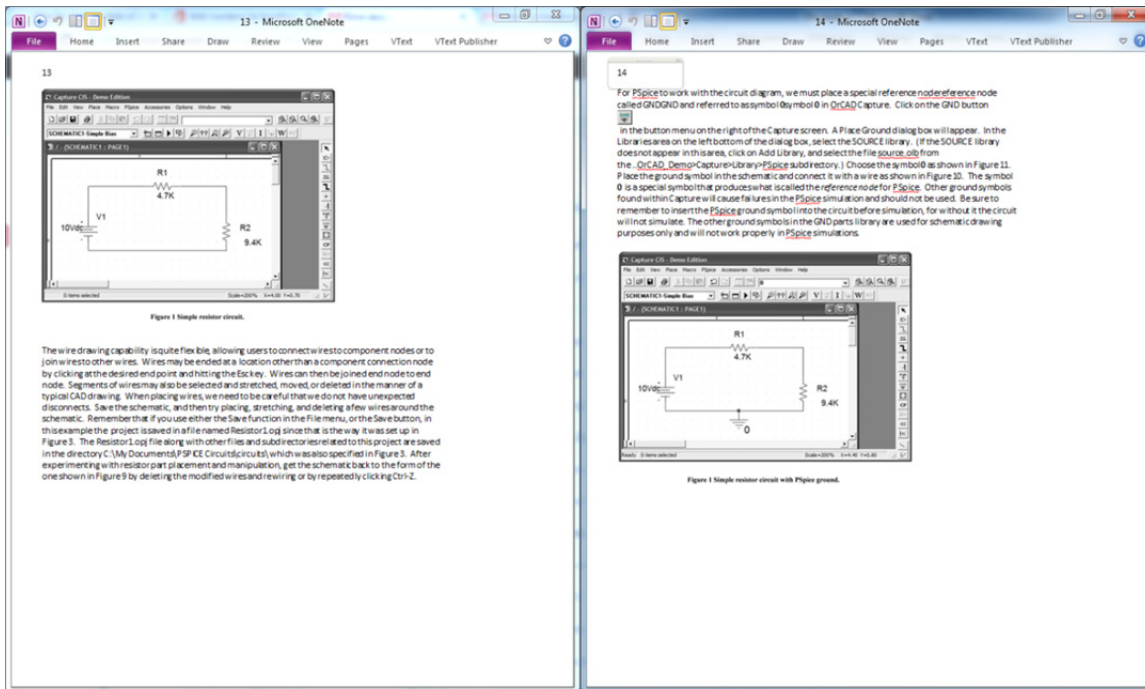


Figure 3.4 Side by Side View (Used with permission from Microsoft [37])

VText also corrects an issue that may come up in modes with two windows. Before inking on a page, the window must have focus (input from the user is directed to the window with focus). Unfortunately, in native OneNote, giving a window focus requires first tapping with the stylus in the window. Often, users will start writing before realizing that the window does not have focus. VText tracks where the pen is hovering, which program is active, and which windows are displayed to give focus to the proper window before the user starts writing. This process may reduce frustration for some users when taking notes.

3.3 Navigation

The standard method for navigating in OneNote is provided by tabs and lists corresponding to the books, sections, and pages as described in Section 3.1 and shown in Figure 3.3. While this allows navigation to any page at any time, it uses a large portion of screen space even when the main action is simply reading. As mentioned in the section Book View, VText can remove the large lists of books and pages to preserve space for content, though navigation can be brought back by the user at any time.

VText adds gestures and optionally a small set of buttons to allow linear navigation. The navigation gestures added allow the user to swipe left and right along the bottom of the screen to move forward and backward, respectively, through the book. Screen areas sensitized for gesture operations are shown in Figure 3.5. While the gestures require a touch PC, VText can also add mouse or pen-sensitive buttons to allow the same linear navigation. These buttons are shown in Figure 3.6. VText's navigational additions allow limited screen space to be used solely for content, rather than navigation. In order to provide feedback to the user indicating that a gesture has been recognized, VText produces, optionally, a recorded sound of a book's page being changed. VText also keeps track of the order pages have been visited. This allows users, by clicking buttons added to the ribbon, to quickly retrace their steps in the book and return to where they were.

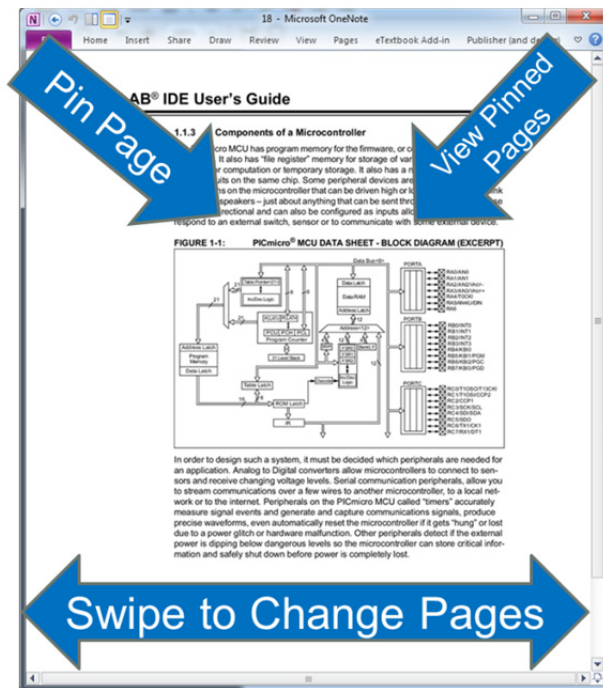


Figure 3.5 VText's Gestures (Used with permission from Microsoft [37])

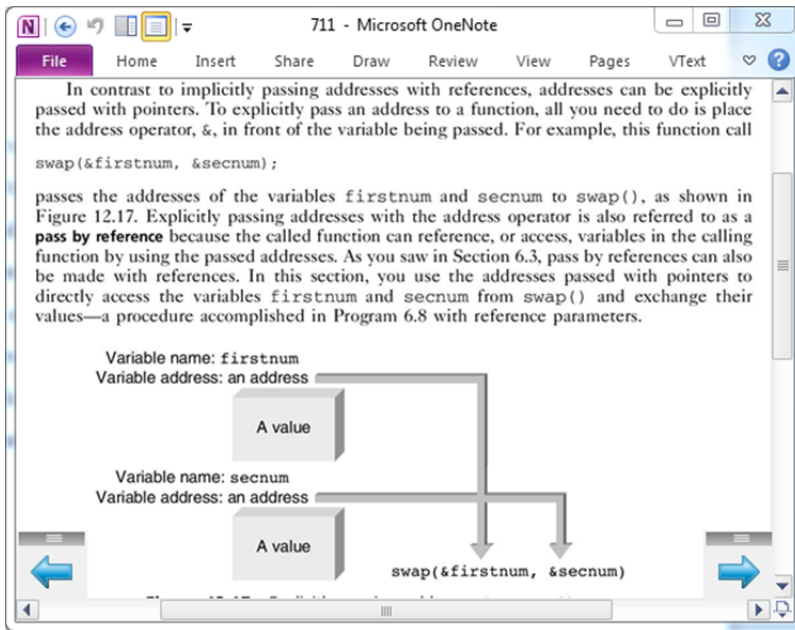


Figure 3.6 Navigation Buttons (Used with permission from Microsoft [37])

Users can also pin pages as a way to bookmark a page so as to easily return to it. Pinned pages can be added or recalled at any time by utilizing the gestures or clicking the buttons added to the ribbon and on the menu which slides up from the navigation buttons shown in Figure 3.7. The pinned pages can be viewed by using the pinned pages view shown in Figure 3.8. The pinned pages view also provides a tab allowing users to navigate to other textbooks without opening the OneNote navigation panes. The pages on this screen are grouped by book and pins from the open book are listed first. This process is designed to provide an improved analog to techniques used by students to bookmark pages in traditional books.

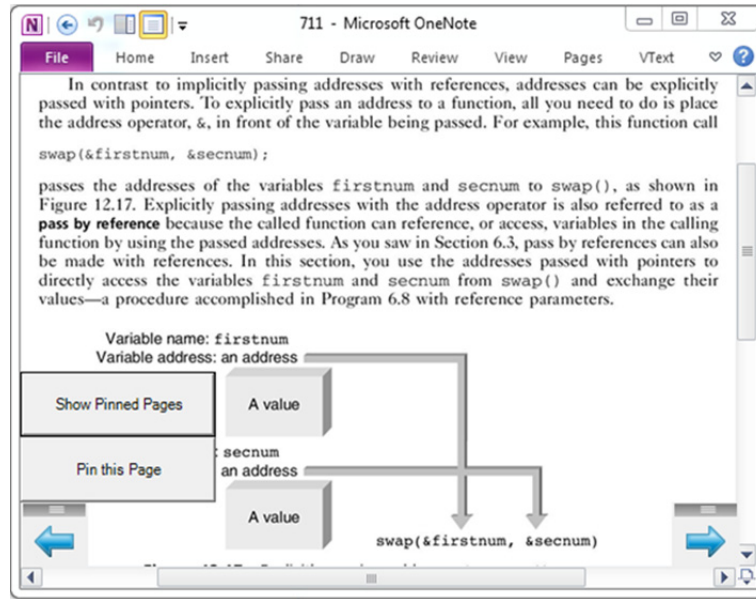


Figure 3.7 The navigation buttons sliding menu with book content in the background (Used with permission from Microsoft [37])

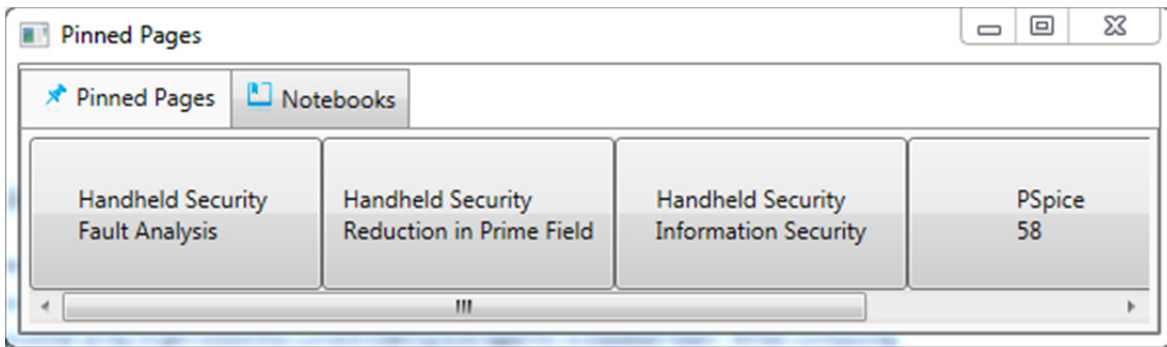


Figure 3.8 The Pinned Pages View (Photo by author 2013)

3.4 Searching the Web

VText’s web search feature allows the user to search the web for any term that is selected. As mentioned in the Active Reading section, the ability to quickly find related documents aids students in exploring subject matter. By facilitating this process, VText allows students to serendipitously discover further information related to course readings. Via the VText Settings menu the user may select which search engine they would like to use. This action is performed by first selecting the text to search, then by selecting “Search <Preferred Browser>” on the right click menu or by clicking the “Search the Web” button on the VText tab of the ribbon menu. The search results open in the default browser as shown in Figure 3.9. The user can return to the

textbook by closing the browser, using alt-tab, or using the Window's taskbar. This feature allows the user to look for other explanations or further reading on a particular topic included in the text or notes material.

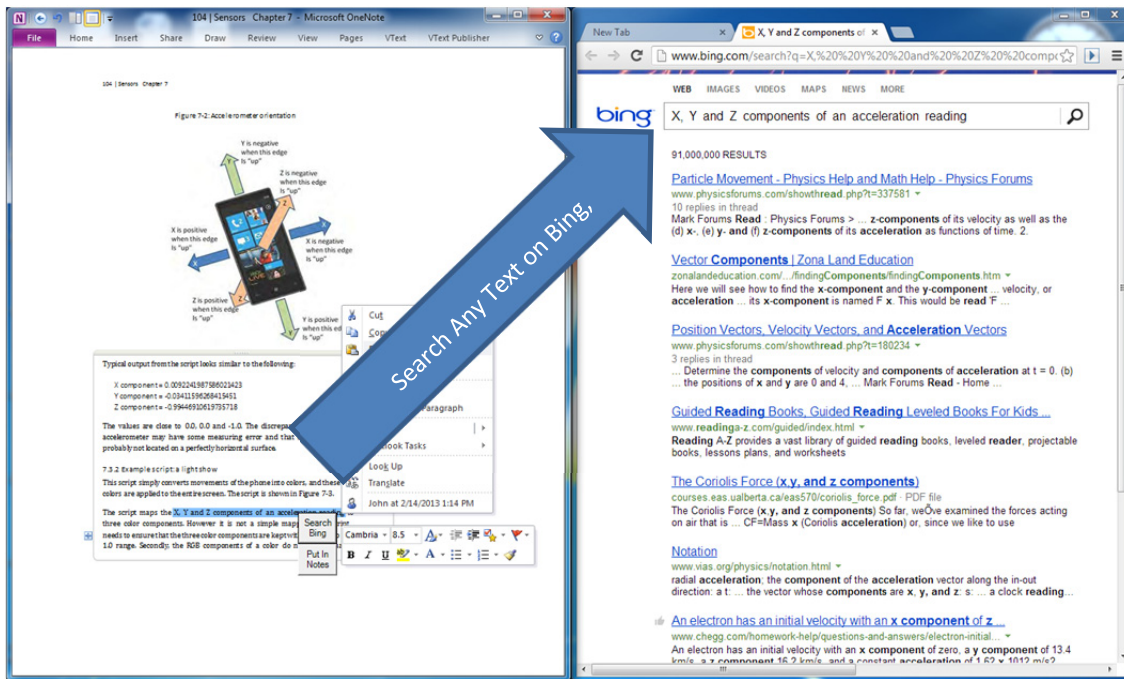


Figure 3.9 VText's Search the web feature (Used with permission from Microsoft [37])

3.5 Notes Page

The notes page allows students to take content or notes from a page in the textbook and store a copy on a dedicated notes page. The user right clicks then selects "Take Note" on the VText menu. The student can later review important clippings from their notes and textbook on the notes page. This feature facilitates students in creating their own summary pages as well as moving annotations they wish to review into one page. Clippings and annotations copied to the page include a hyperlink back to the source page ensuring that important annotations are not lost within the many pages of a book.

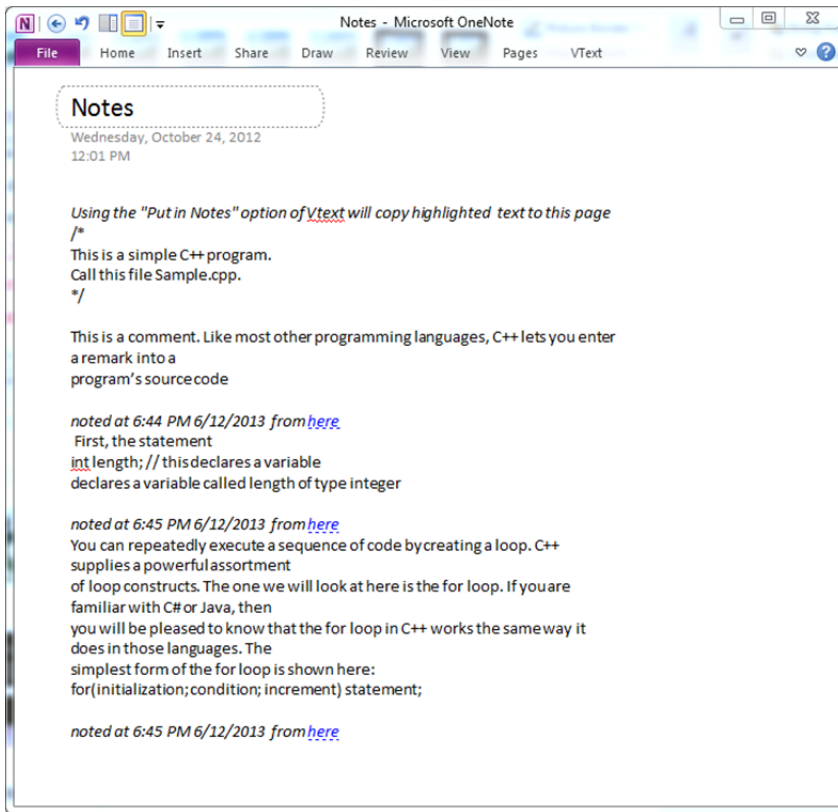


Figure 3.10 A notes page with clippings and links back to the source pages in the textbook (Used with permission from Microsoft [37])

3.6 Classroom Sync

Classroom sync allows instructors and students to stay on the same page during lectures and discussions. A discussion or presentation leader starts a classroom sync session and provides the session identification number to those who wish to follow along. The student's client then listens for the instructor's computer to notify the client of a page change as shown in Figure 3.11. The student's instance of OneNote with VText will then automatically change to the correct page. This feature was developed to address complaints about other electronic textbooks that it took too much time to change between pages during discussions. In order to allow for quick changes back and forth from group and independent work, students may choose to temporarily stop listening and instructors may choose to temporarily stop broadcasting. Classroom sync works well with shared notebooks. This feature should allow discussions and meetings to proceed more easily by reducing the time for everyone to navigate to the same page. This

seamless navigation allows students to focus on lecture materials rather than breaking their thought process to navigate through the course materials.

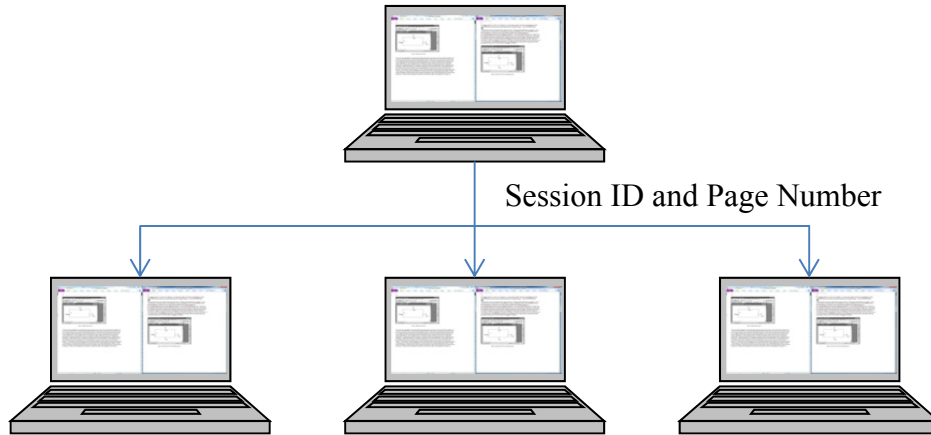


Figure 3.11 Classroom Sync

3.7 Bluetooth Mobile Scanner

The Bluetooth Mobile Scanner feature allows students to use their Android smartphone with cameras to capture images directly into their VText notes. Students simply need to launch the companion Android JBluetooth Scanner application, point it at the document or whiteboard, then click the *take picture* button from their laptop. The picture will be transferred automatically via Bluetooth in just a few seconds. The image is then available to be placed into their notes or onto the clipboard for use in a different program. The concept and the Window's program interface are shown in Figure 3.12.

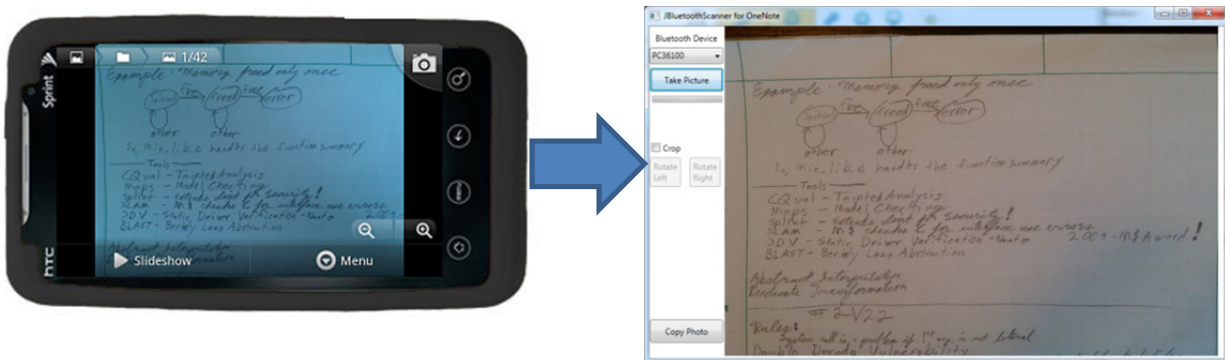


Figure 3.12 The Bluetooth Mobile Scanner including the screen seen on a typical Android phone on the left with an application window containing a captured and transferred image on the right. (Photo by author, 2013)

While the Bluetooth Mobile Scanner can be used to quickly capture images of any type, scanning a quality image of a document lying flat on a desk or table can be difficult due to the awkward position of the camera and preview screen relative to the user. In order to assist the user in lining the camera up properly when attempting to scan documents laying parallel to the ground, the application helps the user determine when the camera's lens is oriented parallel to the face of the document. This reduces skew in the resulting images. The orientation is presented to the user with a familiar carpenter's level interface element using data from the accelerometer. The level element and the Android application are shown in Figure 3.13. On the left, the document does not appear to be skewed matching the level indicators. On the right, the document is skewed due to the device having been held at an angle, matching the level indicator.

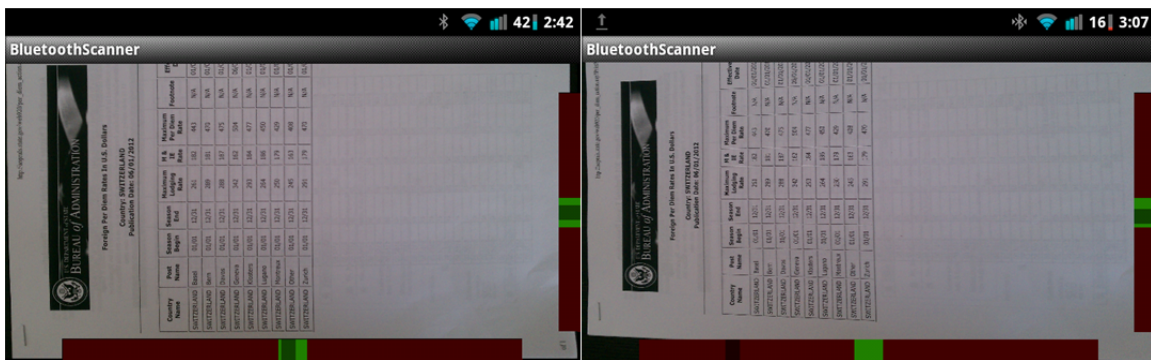


Figure 3.13 Bluetooth Scanner Android Application (Photo by author, 2013)

Figure 3.12 and Figure 3.13 Bluetooth Scanner Android Application show examples of using the Bluetooth Scanner Android Application for capturing notes that include a hand-drawn state diagram and a printed document. Hand-drawn equations, figures, and charts on both paper and chalkboards are other examples of use cases. While equations, charts and other drawings can be created digitally resulting in a potentially cleaner output, these editors are often slow to manipulate the entities which may cause students to be left behind or with incomplete information during lectures. Thus the captured image is left in the form of an original image file inserted into the electronic textbook. This feature is particularly useful for students who do not have pen-enabled laptops or have instructors who use paper handouts and still wish to include these paper documents in their digital notes.

3.8 Content Importers

An important part of encouraging adoption is having content available. To hasten this process, importation tools were created. While OneNote includes the ability to print documents into OneNote notebook format, when this mechanism is used, the entire contents of the document are inserted as a series of images on a single OneNote page. This makes selecting text to search or copy impossible and the reading and navigating of large documents unwieldy as OneNote cannot reflow text in the images to adapt to window size. To remedy this, VText provides tools, the PDF to OneNote (PDFON), PDF to Watermarked OneNote (PDFWON), PDF Rasterized to OneNote (PDFRON), Plain Text to OneNote (TXTON), and Rich Text Format to OneNote (RTFON) importers, to convert popular document formats to OneNote format. Unlike the built-in “Print to OneNote” option, VText imports each page from the original document as one page in OneNote, allowing for easier navigation. To potentially alleviate the intellectual property concerns of publishers, VText also includes an importer that allows publishers to import their content with a watermark. There is also an importer that allows for text documents, a popular format for distributing classic literature novels, to be imported. Each process is completed via the VText Import Wizard which walks the user through importing the appropriate file.

3.8.1 Portable Document Format to OneNote (PDFON)

The PDFON importer takes images and typeset text from PDFs and places them on the OneNote page. The importer makes a best effort determination to maintain spacing of lines and paragraphs of text. Font and text height are also maintained. The PDF is run through a separate companion tool for VText which generates a .onx file. This file contains eXtensible Markup Language (XML) code in a format VText’s importer can easily provide to the OneNote API for import. A sample of output generated by this importer showing that the characters have been regrouped into paragraphs and properly spaced the lines is shown in Figure 3.14.

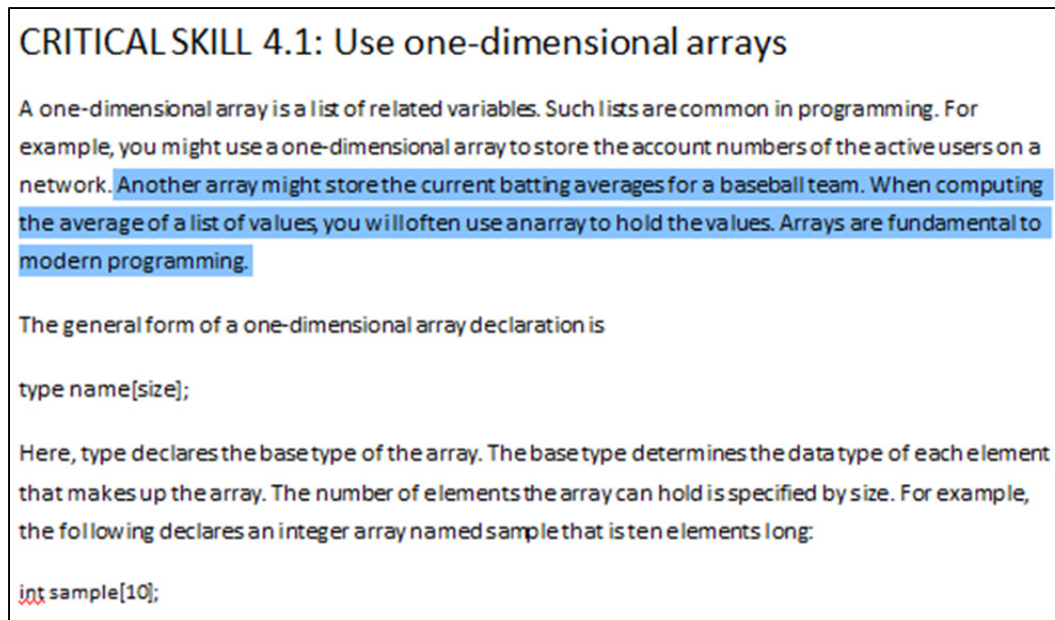


Figure 3.14 Example of PDF Importer with user selected text

3.8.2 Rich Text Format to OneNote (RTFON)

The RTFON importer takes images and text from multi-page RTF documents and places each page on individual OneNote pages. Because of incompatibilities between OneNote and the RTF specification, described below, two problems must be solved requiring the use of a separate solution provided by an included add-in to Microsoft Word. Because the RTF specification does not include information about page demarcation, programs interpreting RTF documents must decide where page breaks should occur based on the layout of the content. A second problem is the inclusion of vector graphics in the RTF specification which OneNote does not support. The add-in uses Microsoft Word's facilities to generate RTF documents which preserve the original content, but are compatible with OneNote. When complete, the files created by the Word exporter can be imported into OneNote for use with VText.

3.8.3 PDF Rasterized to OneNote (PDFRON)

Due to the complexity of PDF documents, full conversion performed by the PDFON converter may not provide desirable results. In order to import these documents into OneNote, each page is rasterized and inserted as a separate page in the VText textbook by the PDFRON importer. While compatible with more documents, this process does not allow the user to select

text in the document which prevents its use with the Searching the Web and Notes Page features. This entire process is completed within OneNote through the VText Import Wizard.

3.8.4 PDF to Watermarked OneNote (PDFWON)

In order to provide digital rights management (DRM) via watermarks. A separate tool available with VText generates a series of images with a unique watermark for each user on each page. In order to generate these files, the source document and a text file containing a list of the desired watermarks are provided to the importer by the distributor. The watermarked images are bundled into a .vti file for each watermark. This file can then be loaded into OneNote via the VText import wizard by each user.

3.8.5 Text to OneNote (TXTON)

The TXTON importer allows plain text format (.txt) documents to be imported. Plain text documents do not contain any formatting information or images. Therefore, the display and layout of the text is left to the program opening the document including if and how to break the text into pages. This importer determines the proper number of words to place on each page to allow for easy reading and navigation in the VText environment. This importer is accessed through the VText import wizard.

3.9 Distribution of Content

Content distribution mechanisms depend upon the import scheme selected. Both fully imported content and intermediary files can be distributed. For content authored in OneNote, the folder including the notebook or the file for the section can be easily distributed as-is through existing infrastructures, including learning management systems. The same options work for all generated file types, ignoring DRM concerns. The main PDF to OneNote (PDFON) importer which maintains content type can be distributed as a OneNote file or as the .onx file. The PDF to watermarked OneNote (PDFWON) importer requires some additional consideration. The external watermarking tool should be run by the distributor, while the importation of the appropriate .vti file should be run by the end user with VText. The distribution of the .vti files requires consideration on the part of the distributor to ensure the correct user and only the correct

user receives the file with the proper watermark. Because the generated files' filename include the unique watermark text, the process can be scripted if the watermark texts map to information identifying the correct account in the chosen distribution system.

3.10 Extensibility and the eQuiz System

The eQuiz System is a feature of VText built to demonstrate how further pedagogically useful extensions are possible within the framework. The eQuiz system is installed alongside VText and consists of a program that allows students to take multiple choice quizzes. The eQuiz System design includes an eQuiz file format (.eqz) which stores the questions and the associated answers and feedback. The .eqz file extension is registered to be opened by the program. These quizzes can be used to provide reinforcement of textbook content.

This .eqz and any other file format, including videos, sound, source code, and simulations, can be embedded into OneNote pages, as shown on Figure 3.16. Double clicking the embedded file will open the appropriate external program. In this case, the quiz taker program will launch asking the student to solve multiple choice questions. This method of conveying extra content can allow for modeling and simulation, assessment, audio visual displays, and tutorials in other programs amongst others. Figure 3.15 shows an embedded C++ file to be used as an example by students reading from a C++ textbook.

To see the `this` pointer in action, examine the following short program:

```

// Use the "this" pointer.

#include <iostream>
using namespace std;

class Test {
    int i;
public:
    void load_i(int val) {
        this->i = val; ← Same as i = val;
    }
    int get_i() {
        return this->i; ← Same as return i;
    }
};

int main()
{
    Test o;

    o.load_i(100);
    cout << o.get_i();

    return 0;
}

```

This program displays the number 100. This example is, of course, trivial, and no one would actually use

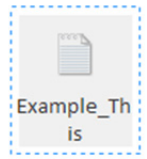



Figure 3.15 An Example of an embedded C++ File in OneNote (Used with permission from Microsoft [37])

The minimum hardware requirements for running PSpice are:

- Intel Pentium 90MHz or equivalent processor
- Windows 95, Windows 98, Windows NT, or greater
- 16 MB RAM (32 MB recommended)
- 90 MB of free hard disk space
- CD-ROM drive
- Mouse or similar pointing device

Review:



Ps spiceElectr
onicsChap...

Figure 3.16 Quiz file included on a OneNote Page (Used with permission from Microsoft [37])

The quiz program displays questions one at a time to the user. The user selects an answer for each question which is stored by the program. Once all questions have been viewed by the user,

the program compares the answers to the correct answers provided in the eQuiz file and can provide feedback. The feedback can be tailored for a specific possible answer. Quizzes can be authored by anyone, including textbook authors, students, and teachers, and the files can be shared embedded in the textbook or as a separate file, for example, by email or via file sharing websites. The file format used to build quizzes is described in Section 4.3.19 . The user interface is shown in Figure 3.17.

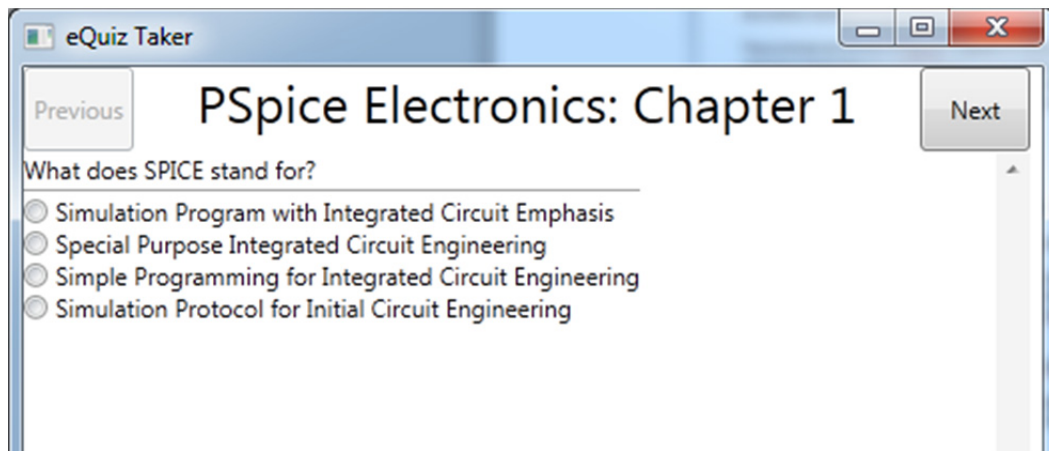


Figure 3.17 The Quiz program (Photo by author, 2013)

Chapter 4

Implementation

4.1 Technologies

The main component of the electronic textbook reader is designed as an add-in to Microsoft OneNote. It is written in C#. The PDF to OneNote XML (PDFON) converter is written in Java. VText also includes a small add-in for Microsoft Word to rasterize vector graphics, also written in C#. The add-ins run with Microsoft Office and are designed for use with Microsoft Windows. The software uses the Word and OneNote Application Programming Interfaces (API), as well as the Win32 API. The main add-in utilizes both the Windows Forms API and the Windows Presentation Framework for displaying custom windows. External libraries were used for handling external file formats and interfacing with Bluetooth hardware. This section provides summaries of the external software utilized, as well as a complete description of the design of the VText software.

4.1.1 .NET

The .NET Framework [33] is a system for executing programs and web applications designed to run on Microsoft Windows and written in the supported languages (C#, F#, Visual Basic). It also provides a foundation for programs written in different supported languages to interact with each other. The .NET Framework includes standard software libraries, known as class libraries, which can be utilized by programs employing the framework. The basis of these operations is the Common Language Runtime (CLR), the interpreter of .NET software for the operating

system. VText uses .NET version 4.0 as it was the most recent version at the time of development.

In the case of Office add-ins, Dynamic-Link Libraries (DLL) provide access to the programs' APIs that allow manipulation of the host programs. While the Office DLLs provide access to any .NET program, add-ins that are designed to operate within the host program are .NET class libraries that implement the IDTExtensibility2 interface. Other add-ins that access content or a feature of the host program do not have to be class libraries. This interface provides methods that allow the host to notify the add-in of startup and shutdown events. During the installation of the add-in, the host program is notified of the new add-in through entries in the Windows Registry. During application program startup, the host program can provide the add-in with various interfaces to modify the host's user interface and actions. This process will be discussed in the context of VText and OneNote later in this chapter.

4.1.2 C#

C# is an object-oriented programming language created by Microsoft to simplify the writing of programs for Microsoft Windows while still providing a high level of control to the programmer [34]. C# provides the most complete interface to the CLR, though it is not by specification required to target the CLR. C# is syntactically similar to Java and C++. It includes a higher level of abstraction than C++ providing the developer with simple facilities for performing complex and common problems, like memory management and graphical output. C# was chosen as it was the strongest and most familiar of the languages available for add-in development.

4.1.3 Java

Java is an object-oriented programming language that can be run on any platform for which a Java virtual machine has been created [35]. To meet its goal of providing a language that is both simple for new programmers and familiar to experienced programmers, syntax similar to C++ was chosen while abstracting some complex processes from the developer, like C#. Also, the standard libraries provide a wide range of capabilities to the programmer. Java was used for a small portion of this project which required the iText library, a PDF interpreter, which was most

easily accessed through its Java API. It was also used for the development of the Android application for the Bluetooth Mobile Scanner.

4.1.4 XML

Extensible Markup Language (XML) is a versatile file format consisting of Unicode characters, allowing it to be human and machine readable. Its design allows for use with many different applications which require the transfer of data. The format is suitable for use with object oriented programming as data types, called nodes, can consist of primitive data types as well as other classes. This data can take many forms. For instance, binary image data can be encoded in Base64 to allow for storage in XML. The versatility of XML allows for common code to be used to interpret many different types of data. Applications using XML can specify a certain schema. A schema is a set of rules indicating which nodes, attributes, and sub-nodes can be included or are required in the file. For instance, VText uses XML for intermediary document storage as well as user preferences, while OneNote's API uses XML to provide information about content stored in the program. Each category of data uses a different schema.

4.1.5 Windows Presentation Foundation

The Windows Presentation Foundation (WPF), a part of the .NET Framework, allows for the creation of graphical interfaces that are coded using Extensible Application Markup Language (XAML) and can be manipulated in graphic designers, which allow drag and drop user interface design [36]. WPF utilizes DirectX for rendering rather than the older Graphics Device Interface (GDI) used by Windows Forms. The WPF XAML allows the programmer to separate the interface coding from the logical coding.

4.1.6 OneNote API

The OneNote API provides other programs with access to its functions and content. [37]. The API is broken into three categories the Window Interface, the Quick Filing Dialog Box Interface, and the Application Interface. The Window Interface allows the manipulation of OneNote windows currently open. The Quick Filing Dialog Box Interface creates a user interface element to allow the selection of notebooks and pages. The Application Interface

provides access to the content and allows control of page navigation. VText uses the Window and Application interfaces.

The Window Interface allows control of the OneNote windows. It can be used to create *side notes*, which are small windows displaying pages from a side note section, and *docked notes* which are windows open on the side of the screen and always visible. This interface also allows manipulation of the user interface. Using the Windows Interface, the ribbon and the lists of notebooks, sections, and pages can be hidden to leave more room for the content. Finally, this interface also exposes the window handle which can be used for further manipulation of the OneNote window(s) through the Win32 API.

The Application Interface provides many functions for interacting with OneNote. This interface can provide XML data containing a list of all notebooks, sections of notebooks, and pages the user has created. The XML includes information about each page, section, and notebook's last modified time, ID number, name, and other data. This information can be used to generate a list of notebooks and pages a user can navigate to in OneNote. The interface also allows an individual page's content to be viewed and edited via XML. The ability to edit pages programmatically allows the add-in to manipulate, insert, and parse content. Accessing the page content enables the add-in to determine what text a user has selected, if any, on the page. The Application Interface also allows the creation of new pages. These functions are used extensively by the content importers of the add-in. The interface also exposes control of window navigation. By utilizing the list of pages and this function, a program can choose which page to navigate to based on its desired operation. VText uses this for the capabilities described in Navigation as OneNote does not provide a simple method to navigate linearly through a textbook, particularly when a window must skip a page when two windows are displayed in sync.

4.1.7 Android

Android is an operating system designed for mobile devices [38]. The devices which run Android and its competitors have achieved widespread usage in past years. These devices tend to have many sensors and radios for data collection and information transmission. The ubiquity of these devices makes them a reliable tool for non-critical applications i.e., those applications

that aren't necessarily secure, could be lost with no significant detrimental effect, etc. Programs are generally created by extending the Android Activity class which contains functions the operating system will call when the program is launched. During these calls, the program can set up a user interface, register to receive events from the various sensors, amongst many other possibilities. VText utilizes the camera and Bluetooth radio to provide an always-available scanning and photo capture system. Android programs interact with the operating system in an event driven manner. These events range from user interface actions to sensor status changes. The Android operating system and API allow for rapid development of applications.

To develop Android applications, a developer uses the Android Software Development Kit (SDK). The SDK includes software build tools which compile Java source code into an Android Package File (APK). While a traditional command line interface can be used for designing, writing, compiling, and testing code, the Eclipse Integrated Design Environment (IDE) with the Android add-ins (called plug-ins by Eclipse) simplifies these processes.

4.1.8 Visual Studio

The development of VText was performed in Microsoft's Visual Studio 2010. Visual Studio is an IDE which provides the developer with tools to simplify the design process of programs intended to run with Microsoft technologies, such as .NET, on Windows computers. The tools provided by the IDE help the developer configure projects, write code, document code, design user interfaces, and compile programs for distribution. Due to the many libraries and components included in VText, manual configuration of building the final executable would have been complicated, but Visual Studio handles this process automatically.

4.2 Libraries

Libraries allow code to be reused quickly and easily. VText utilizes a number of libraries in order to hasten development. For example, parsing of PDF documents is handled by libraries to reduce the complexity of VText. Other libraries, such as Win32, allow the add-in to perform actions only the operating system can execute including manipulating the windows of external programs and launching other programs. More detail will be provided throughout this section.

4.2.1 Win32

Win32 provides access to many of the functions Windows can perform. This library exposes operations handling processes, threads, controls, and graphics. The user32.dll is used to manage information pertaining to individual windows. By using the window handle provided by the OneNote API, OneNote windows can be manipulated in size and location. Other useful information about windows open on the system can be determined such as if the window is visible, if it is in focus, what screen space it is using, and which process owns it. This library also allows for hooking into low level events produced by hardware. Hooking into an operating system's low level events asks for indication when a particular set of events has occurred. For instance, implementing gestures can require low level information about mouse, pen, and touch actions. These events are usually handled by the window in which the event occurs. This can be a problem if the event will be managed by a window owned by an application which is different from the programmer's application. Separate programs, such as add-ins, rely on the host application to relay information. In the case of OneNote add-ins, these events are not relayed requiring the add-in to ask the operating system directly for mouse, pen, and touch event notifications. This library also provides the ability to send messages to windows. These messages can include asking them to close, maximize, take focus, and other state change events. The Win32 library is critical for applications that wish to interact with the operating system.

4.2.2 Ghostscript and GhostscriptSharp

Ghostscript is a software package, available under the Affero General Public License, designed to interpret and rasterize PostScript and PDF documents [39]. This capability can be used to transform these documents into bitmap images which can be used in programs which do not directly support PDF. GhostscriptSharp allows programs written in C# to interact with Ghostscript capabilities through a wrapper class [40]. VText uses GhostscriptSharp for the PDFRON importer to rasterize the source PDF into images to be inserted on OneNote pages.

4.2.3 XDesigner RTF

The XDesigner RTF DOM Parser, available under the GNU2 license, allows the interpretation of Rich Text Format (RTF) documents in a C# environment [41]. RTF is a

common format for saving text with layout and formatting information. By using this library, the importation of RTF documents can be supported. The RTFON importer utilizes the RTF DOM Parser to extract text, images, and format information from RTF documents in order to translate these documents into OneNote format.

4.2.4 iText

iText is a PDF interpretation and creation library available under the Affero General Public License [42]. It allows PDF documents to be parsed in an event driven manner. Because PDF files are not designed to be edited, the specification does not require that information about how characters are grouped together, such as words and paragraphs, be stored. Because of this, iText simply parses the document and notifies the implementing program as characters and images are parsed. Unfortunately, the library does not regroup these individual characters and groups of characters into a complete document, but only as the information is stored, usually as individually typeset characters. However, it allows text to be preserved as text, which cannot be done when simply rasterizing. iText performs the extraction of text, image, and format information from PDF documents so that they can be translated by the PDFON importer into OneNote format.

4.2.5 DotNetZip

DotNetZip is a compressed archive tool designed for use in .NET applications [43]. It allows the creation of standard zip archives containing many files. It fulfills the need for a multiple file archive format which the standard .NET libraries do not include. By using the zip format, many files can be transferred easily as a single file while the compression can reduce the file size. The PDFWON importer uses DotNetZip to unzip the .vti files when importing the watermarked images into OneNote. While the compression adds little benefit when archiving images, the archive allows a single file containing many images, one per page, to be distributed to each student for each book.

4.2.6 32feet.NET

The 32feet.NET library, available under a custom free license from the library's developers, allows Windows computers to communicate on personal area networks [44]. While personal

area networks include many technologies, this project utilized the Bluetooth capabilities. The library allows for Radio Frequency Communications (RFCOMM) and device discovery. RFCOMM emulates RS-232 serial communication allowing for general purpose communication. To simplify this process, Bluetooth includes the Serial Port Profile (SPP). SPP allows applications to abstract the details of a virtual serial port and simply use the connection as a data stream. As a standard Bluetooth profile, RFCOMM/SPP is available in most Bluetooth implementations, including Android. Device discovery allows the host computer to discover Bluetooth devices that have previously been paired with the device as well as new devices whose Bluetooth radios are set to visible. The Bluetooth Mobile Scanner uses this library to enable the Windows computer to receive the scanned image wirelessly from the Android device.

4.3 VText Components

VText is designed to be modular such that each class is responsible only for particular categories of tasks. User interface code is separated from logical code as often as possible following standard design practices. This section describes the design and operation of the software components of VText individually.

4.3.1 Initialization & Shutdown

OneNote notifies the add-in of events in the program lifecycle by calling methods included in the `IDTExtensibility2` interface. In order to implement the interface, VText includes functions `OnConnection` and `OnDisconnection` which notify VText that the program is starting and closing, respectively. To produce custom ribbon controls, the `IRibbonExtensibility` interface must be implemented. This requires the inclusion of a `getCustomUI` function which provides the XML for the ribbon. The host application parses this file and adds the controls. To include icons, the XML includes the function name of the function in the implementing class to call to get the icon for each custom ribbon control.

The loading process begins with the host application calling the `OnConnection` function. In this function VText loads the settings, pinned pages, and page cache from previous executions. Then, low level mouse hooks are inserted and the system thread is started. OneNote

proceeds by calling the `getCustomUI` and the specified function to get the controls' icons. From this point, all operations, except the system thread, are event driven.

When notified through the `OnDisconnection` function that the program is closing, `VText` saves the page cache, pinned pages, and settings; closes any open windows; ends all spawned threads; and removes the low level mouse hooks. Once the function returns, `OneNote` proceeds with its own shutdown process. In case the system crashes while using `VText`, the user's information will be safe. `OneNote` continuously saves changes to notebook content while `VText` can gracefully recover from the loss of information. A high level view of the control flow is presented in Figure 4.1.

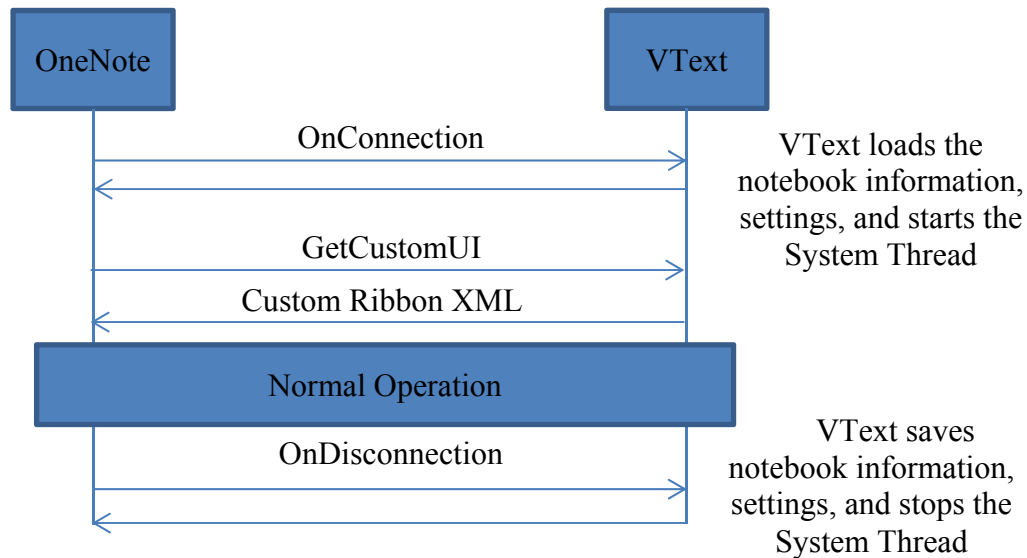


Figure 4.1 VText's Initialization and Shutdown Process

4.3.2 System Thread

Unfortunately, the `OneNote` API does not provide notifications to fully enable event based programming for the add-in. Because of this, the add-in spawns a system thread which runs every half second via a timer to perform actions which cannot rely on events. This thread checks if pages have been added or deleted (see Page Cache), if windows were moved or resized (see

Window Manipulation), if the stylus was moved to a new window (see Window Manipulation), or if the currently displayed page has changed (see Navigation).

The thread is implemented as an `ElapsedEventHandler` utilizing the .NET `Timers` library. Using scheduled thread execution allows the thread to release all resources between runs while running at a reliable interval.

4.3.3 Ribbon

Microsoft Office products since the 2007 edition have included an interface element known as the ribbon which is a part of the Fluent UI. The ribbon attempts to help users more easily locate tools they need by grouping similar actions in categories such as “Insert” and “Page Layout.” Office add-ins may add custom controls to the ribbon to give users access to the added functionality. To add controls to this interface, add-ins must implement the `IRibbonExtensibility` interface. This interface includes one method which asks the add-in for XML data specifying the controls the add-in wishes to add to the ribbon.

The XML is written using the Office Fluent UI schema. This schema includes elements for specifying tabs, groups, and controls for the ribbon. The top level element, `customUI`, includes the attribute `loadImage` which tells the host which function in the class extending the `IDTExtensibility2` interface to call to load icons for the controls. The `customUI`'s child node is a `ribbon` which can consist of as many `tabs` as needed. Tabs, which will all be shown on the screen at one time, are then split into any number of groups to further organize the controls. Groups then include any number of controls such as `buttons`, `togglebuttons`, and `checkboxes`. The control elements include attributes that identify which icon should be provided by the function specified by `loadImage` and which function should be executed when an action is performed on the control. Other attributes provide information about the controls such as the text label and size. Code Snippet 1 includes an excerpt of the XML code used in VText including the ribbon, tab, group, and user interface element tags and attributes used to define the custom ribbon.

Code Snippet 1

```
<?xml version="1.0" encoding="utf-8" ?>
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"
loadImage="OnGetImage" onLoad="ribbonLoaded">
  <ribbon>
    <tabs>
      <!-- Our content goes here... -->
      <tab id="Views" label="VText">
        <group id="start" label="Views">
          <splitButton id="twowindowmode" size="large">
            <button id="sidebyside" label="Side by Side" image="bookicon"/>
            <menu>
              <button id="bottombytop" label="Vertically Tiled"
onAction="switchViewMode"/>
              <button id="homework" label="Homework" screentip="Opens the same page side
by side" onAction="switchViewMode"/>
            </menu>
          </splitButton>
          <button id="onepage" label="One Page" size="large" image="notebook"/>

          <toggleButton id="supplementalnavigation" label="Navigational Buttons"
size="large" image="compass" onAction="supplementalnavigation"
getPressed="checkSupplementalNavigation" />
        </group>
      </tab>
    </tabs>
  </customUI>
</ribbon>
```

4.3.4 Window Manipulation

VText manipulates the windows to display content in a more useful way to the reader using both the OneNote and Win32 APIs. The main feature for improving content display involves moving OneNote windows to display subject matter in an efficient manner. Through the OneNote API, the window handles for the OneNote program can be obtained. Window handles are references to the window in memory which provide the Win32 API with the necessary information to manipulate the size and location of the OneNote windows.

When a user first starts OneNote a single window is opened. If the user clicks the control to open a second window side-by-side with the first, a second window is opened by launching a new OneNote process. VText's `WindowManager` is used to perform this action. In order to determine the proper sizes and locations of the windows, the Win32 API is used to determine the screen size. The Win32 API is used again to move and resize the two windows into the left and right side of the screen. The OneNote API can then be used to remove OneNote's interface for

navigation if requested. The OneNote API and the Page Cache, described in Section 4.3.5, can be used to navigate the second window to the subsequent page if the “synchronized pages” setting is set. A similar process is used to switch to the other multi-page modes described in Book View.

VText also manipulates which window is in focus to ease writing in different windows as described in Book View. As part of the main thread, VText utilizes the `FocusManager` to check the mouse or pen location to determine which window it is hovering over. The `FocusManager` takes the mouse location, and determines if the mouse is hovering over a OneNote window. If so, it uses the Win32 API to give the window focus. A window having focus allows it to receive input such as pen and keyboard and be completely in view. However, before it does this, it must make sure that there is not a different window, possibly from another process, obscuring the OneNote window indicating that the user is likely still using the other program.

Checking for obscuring windows is implemented in the `WindowManager` and requires considering all of the system’s open windows in front of the OneNote window. Each of these windows must be checked to determine if they are overlapping the OneNote window and visible. To find windows in front of a particular window, the `getWindow` function of the Win32 API can be used. Windows uses the z coordinate of a three dimensional coordinate system to determine the order of drawing windows diagramed in Figure 4.2 Diagram of Window Layout including Z-Order. Windows that are higher in the z-order are drawn on top of those lower in z-order. By traversing the list of windows in decreasing z-order, it can be determined which windows are in front of other windows. This allows the `WindowManager` to determine if a OneNote window is obscured or not.

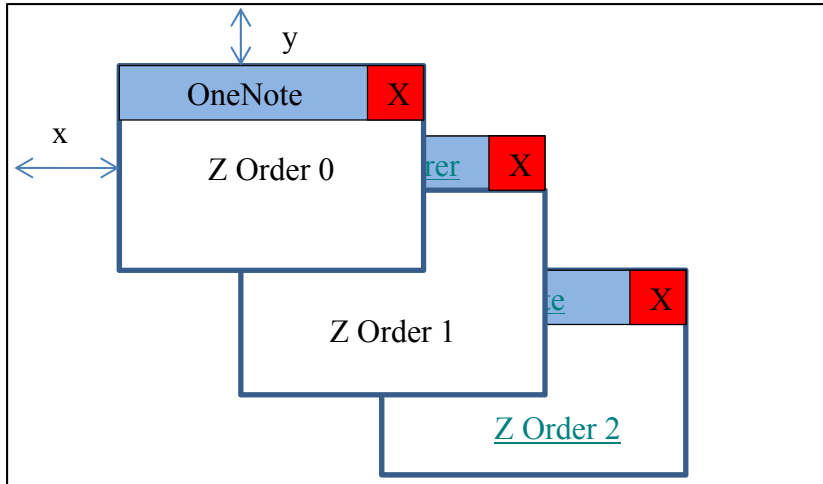


Figure 4.2 Diagram of Window Layout including Z-Order

The `WindowManager` can also determine when a window has moved or been resized as well as if a particular coordinate is inside the bounds of a OneNote window. Both use the Win32 API to determine the size of the OneNote windows. To determine if the window has moved or been resized the `WindowManager` compares the current size and location with previous data stored by `VText`. To determine which window a coordinate is in, it compares the coordinate to the bounds of each OneNote window as well as using the function described above to determine if the coordinate is contained in an overlapping window. These functions are utilized by other parts of the add-in.

4.3.5 Page Cache

Although the OneNote API provides a list of pages in XML format, finding the next page to navigate to on a user's command is too slow. Unfortunately, there is no other method in the API to provide this information more quickly. Therefore, `VText` keeps a cache of all of the pages in the user's notebooks. The process of building this cache is performed the first time `VText` is run. Subsequent changes are found during an iteration of the system thread. To reduce the burden of rebuilding the cache multiple times, the XML is hashed which takes far less time than the parsing. Only if the hash value has changed is the cache rebuilt.

The OneNote API exposes the page ID for the currently viewed page. In order to provide quick access to the next page, the currently viewed page's ID is used to locate the current page's node in the cache utilizing a hash table. Particularly, this is implemented using .NET's

Dictionary class. The nodes also maintain the previous and next pages as members of each element in the cache. This data structure behaves similarly to Java's linked hash map for which there is no direct C#/.NET equivalent. This data structure is shown in Figure 4.3 below. A separate dictionary is stored to allow look ups based on the name and creation time of the page for use with the Classroom Sync feature; however, the page objects are the same to conserve memory. The page objects contain information about the page that the add-in needs frequently and quickly.

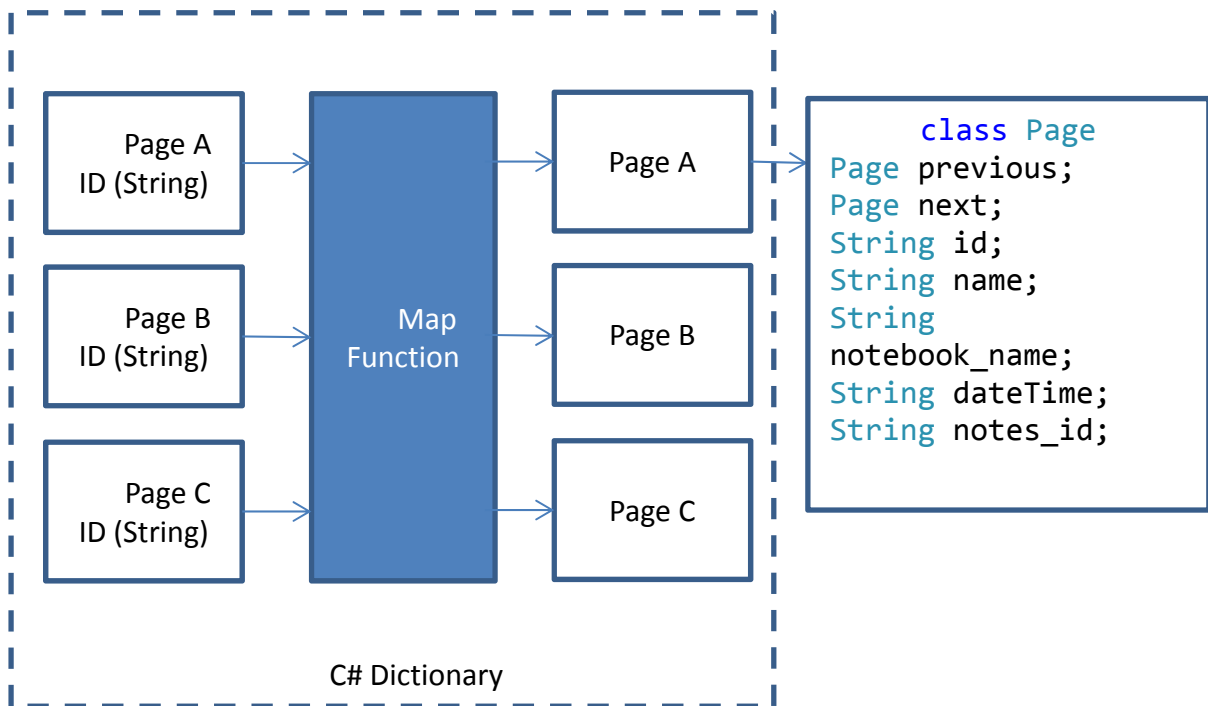


Figure 4.3 The VText Page Cache

The cache is saved by VText in a file of its own format so that it can be quickly rebuilt when the program is restarted. This file simply includes the data VText is interested in and can therefore be parsed more quickly than the XML provided by OneNote.

4.3.6 Navigation

While OneNote provides navigation mechanisms allowing users to find pages based on notebook and section, VText adds capabilities to allow for navigation without utilizing the full OneNote UI. VText can add navigation buttons which are simple `Forms` created through the Windows Forms API which contain a single button. When the system thread finds that the

windows have been resized, the buttons are automatically moved to the new location as shown in Figure 4.4. They also react properly to windows being opened and closed, moved, or synchronized by utilizing functionality of the `WindowManager` described in Window Manipulation. The proper location is calculated by finding the windows' positions on the screen through the Win32 API. When clicked, the page cache is utilized to look up the page that should be navigated to. The buttons are also set up so that if a user clicks, or touches, and drags upwards, a new menu appears allowing the user to pin the current page or to view all pinned pages. The window is created using Windows Forms. These actions can also be performed by gestures (see Gestures).

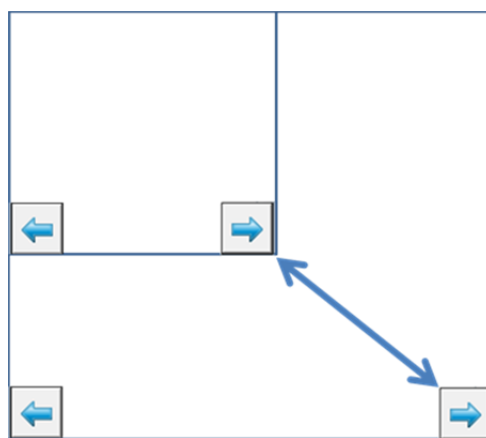


Figure 4.4 Navigation Buttons Automatically Adjusting Position

Pinned, or bookmarked, pages are handled through the `PinnedPageManager`. The `PinnedPageManager` handles pinning pages, storing the pinned pages, and displaying the pinned page view. The pin page operation can be triggered through a gesture, a ribbon button, and the navigation buttons. To save the pinned page, the `PinnedPageManager` adds the pages ID to an `ArrayList`. When removed through the pinned page view, it simply removes the ID from this list. When OneNote is shut down, the pinned pages' IDs, which are strings, are saved in a simple line separated file to be read back in when restarted.

When the pinned page view is requested, the `PinnedPageManager` opens a Windows Presentation Framework (WPF) window showing buttons for all of the pinned pages. This view also has a separate tab that shows all of the notebooks so that other books can be easily navigated to. This window uses the WPF because the graphical user interface description, specified in

eXtensible Application Markup Language (XAML), is complicated enough to have value in being defined separately from the logical code. A portion of the XAML is shown in Code Snippet 2 below. By using data binding, a list of buttons can be generated simply by providing an `ArrayList`, in this case of notebooks or pinned pages. The WPF automatically creates the list of buttons based upon the list provided in the code for the `PinnedPageFields` variable. The bindings are specified by including the name of the `ArrayList` in the `ItemsSource` attribute of the `ItemsControl` element. The `ArrayList`'s elements include the name of the page or notebook, which is bound to the `Button`'s `Content` attribute, and the page or notebook ID, which is bound to the `Button`'s `Tag` attribute. When the user clicks on a button, the function specified by the `Click` attribute is executed. The `Tag` can be read by this function and is used to direct OneNote to navigate to the page.

Code Snippet 2 Excerpt of XAML to define one tab of the Pinned Pages View

```
<Window mc:Ignorable="d" Height="175" Width="600" Title="Pinned Pages">
  <TabControl>
    <TabItem Height="30"><!-- Header="Pinned Pages" Height="25"-->
      <TabItem.Header>
        <StackPanel Orientation="Horizontal">
          <Image Height="18" Source="..\resources/pin.png" />
          <TextBlock Text="Pinned Pages" Margin="2,0,0,0"
VerticalAlignment="Center" />
        </StackPanel>
      </TabItem.Header>
      <DockPanel Name="pinnedPagesSP">
        <ScrollViewer Name="scrollviewer"
          VerticalScrollBarVisibility="Auto"
          HorizontalScrollBarVisibility="Auto"

          PanningMode="HorizontalOnly">
          <Grid>
            <ItemsControl ItemsSource="{Binding
Path=PinnedPageFields}" Name="theItemsControl">
              <ItemsControl.ItemsPanel>
                <ItemsPanelTemplate>
                  <StackPanel Orientation="Horizontal"/>
                </ItemsPanelTemplate>
              </ItemsControl.ItemsPanel>
              <ItemsControl.ItemTemplate>
                <DataTemplate>
                  <Button Content="{Binding Path=Name}"
Tag="{Binding Path=ID}" Width="150" Click="Window_Click"
MouseRightButtonDown="Button_MouseRightButtonDown" />
                </DataTemplate>
              </ItemsControl.ItemTemplate>
            </ItemsControl>
          </Grid>
        </ScrollViewer>
      </DockPanel>
    </TabItem>
  </TabControl>
</Window>
```

VText also tracks a list of previously visited pages so that users can easily return to pages they were recently reading. The system thread recognizes the page has changed and adds the new page to the list. On request, a list pointer moves back one position and navigates the user to the proper page, similar to popping an element off of a stack. The user can then move forward in the stack or go elsewhere creating a new path.

4.3.7 Gestures

Adding gestures to OneNote proved to be a challenge as OneNote does not provide add-ins with any indication of mouse events. While the add-in does occasionally poll the mouse to

determine which window the mouse is hovering, this could not be used to implement gestures. The mechanism used by the `WindowManager` does not provide access to the status of the mouse buttons nor could it be reasonably accessed at the necessary rate. The `WindowManager` technique also cannot differentiate between mouse, pen, and touch. Therefore, the add-in utilizes `Window`'s low level hooks.

Low level hooks allow a program to be placed into the chain of programs notified of raw mouse, pen, and touch events by the operating system. This requires the programs that are inserted in this chain to quickly handle the event and return so that the next program is notified in time to perform its operations. `VText` performs recognition on these events in a separate thread so that it will not disrupt the event being passed to other programs. By the nature of the hooks, `VText` receives information about all mouse events on the computer. Therefore, it must determine which events are relevant. To do this it uses a function of the `WindowManager` to determine if the mouse event is in a OneNote window. Once this is determined, relevant events are considered for gesture recognition and opening the right click menu. Right click actions located within a OneNote page trigger the `VText` right click menu to open. Further, the input device must be classified as mouse, pen, or touch as the user specifies which devices trigger gestures.

The gestures recognized by `VText` are simple linear movements in particular areas of the `VText` window. Upon recognizing a gesture the relevant manager is called to perform the necessary task. Unfortunately, these touch events cannot be blocked from processing by OneNote. Therefore, these gestures are still treated as panning events by OneNote causing the page to be shifted before leaving the page. To rectify this problem, `VText` sends a series of mouse events which cause OneNote to pan back in the other direction cancelling out the panning performed during the `VText` gesture.

4.3.8 Settings

`VText` allows users to select preferences about the operations of the program. In order to manipulate these settings, a WPF window is launched. The settings are then changed in the `SettingsManager`. Upon the closing of OneNote, the settings are saved in an XML format

document. A sample settings file is show in Code Snippet 3 below. Upon restart, the settings are reloaded into the `SettingsManager` from this file.

Code Snippet 3 Sample Settings File

```
<settings>
  <searchengine>0</searchengine>
  <sound>true</sound>
  <touch_gesture>true</touch_gesture>
  <pen_gesture>false</pen_gesture>
  <mouse_gesture>false</mouse_gesture>
  <rcm_open_direction>false</rcm_open_direction>
  <group_pins_by_notebook>true</group_pins_by_notebook>
</settings>
```

In the settings file, the user's preferred search engine is saved as an integer which corresponds to a list defined in the code. Sound is simply a Boolean specifying if VText should make noises for page changes and pins. Touch gesture, pen gesture, and mouse gesture are Booleans noting which input devices should trigger gestures. Right-click menu open direction is a Boolean specifying if VText's additional right-click menu should open left or right of the click event. Group pins by notebook stores if the pinned page view should group pinned pages by notebook or not.

4.3.9 Right Click Menu

To allow for more options in context related situations, VText has its own right click menu. When a user right clicks, or long presses for pen and touch, OneNote has a context menu appear. This event is recognized by the low level hooks otherwise utilized for gestures. VText adds another context menu allowing the user to search the web and add selected text to the notes page. In order for this window to not overlap the OneNote menus, VText checks the value of the `HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\MenuDropAlignment` key in the Windows Registry. This key specifies if these menus should open to the left or the right of the mouse click. VText automatically opens to the side opposite that of the normal behavior so that the new menu will not be obscured. This can be overridden by the user on the settings menu if necessary.

4.3.10 Web Search

The web search feature is provided by allowing users to select words then clicking on either the button on the right-click menu or the ribbon to search the web. To perform the action, the add-in utilizes the OneNote API to access the current page and its content. The XML representation of the text elements includes a `selected` attribute. Once the selected text is found, the add-in starts a new process with the web address of the selected search engine including the search parameters. Windows automatically resolves this to opening the default browser with the specified address.

4.3.11 Notes Page

Sending text to the notes page uses the same mechanism as web search. The selected text is taken from one OneNote page and automatically inserted in the notes page for the current book, both using the OneNote API. The notes page is simply a page in the book with name “Notes” and should be the first page in a book or section.

4.3.12 Bluetooth Mobile Scanner

The Bluetooth Mobile Scanner utilizes software on the host computer as well as on an Android device. The communication with the 32Feet.Net Bluetooth library was designed to be a dynamically linked library (DLL) so that the same code can easily be reused by other add-ins or programs. This middleware library is called `BluetoothImageCom`.

When the Bluetooth Mobile Scanner program first starts, the client computer scans for devices it has previously paired with and other devices which have their Bluetooth radio set to visible. The user selects the device from a drop down menu then clicks a button on the interface to take a picture. This action begins the acquisition process. The `BluetoothImageCom` library sends a “take picture” command through the 32Feet.Net library which opens two data streams, one inbound and one outbound, between the client and Android device. The client then waits for the Android device to respond. Once the Android device has taken an image (this process is detailed later in the section) it will write the data to the inbound data stream opened by the client. The `BluetoothImageCom` library then reads from the inbound data stream to reassemble the image which it returns to the client application, in this case the add-in. Figure 4.5

shows the process to acquire a picture. Note that only the software represented by the gray box labeled 32Feet.NET library was developed by a third party.

Once the image is received by the client computer, the image is displayed to the user. The user can then rotate and crop the selected image through functions provided by .NET. The image can then be copied to the clipboard. This is accomplished through the `Clipboard.NET` class library which exposes the system clipboard.

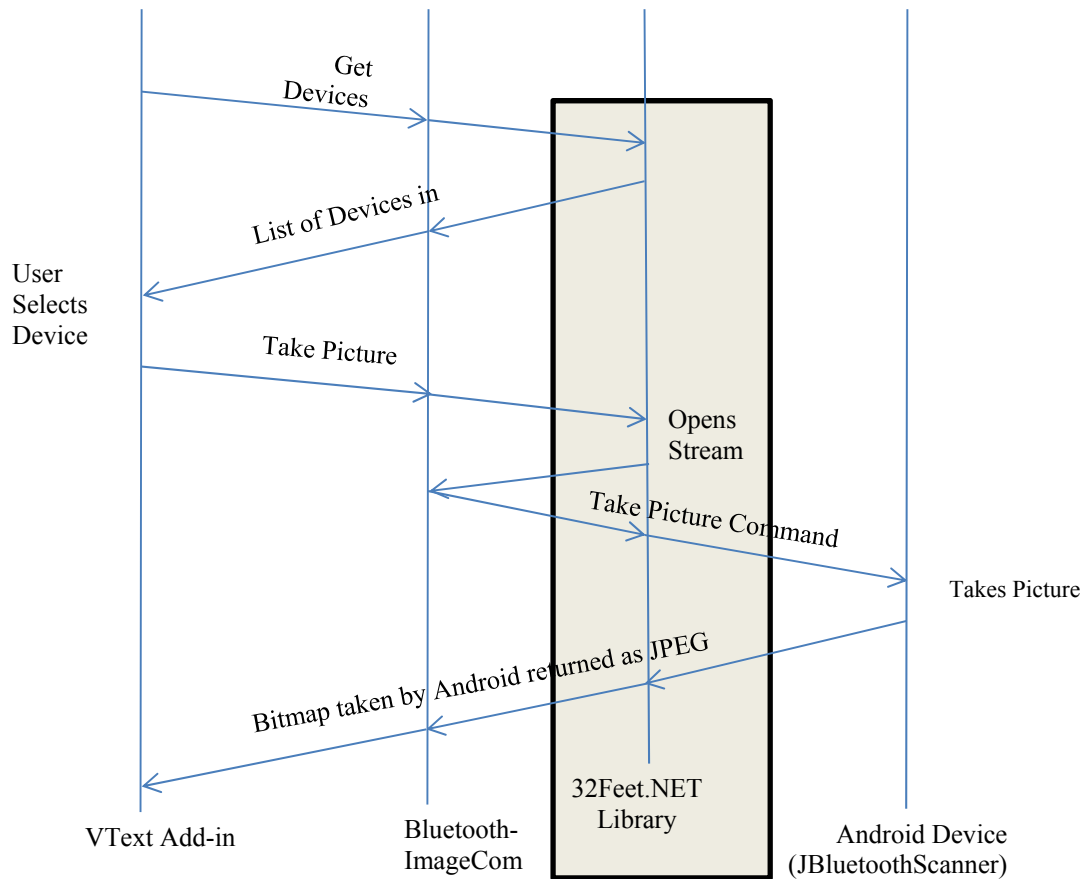


Figure 4.5 Bluetooth Scanner Communications Outline

The Android program is designed to work with any program utilizing the `BluetoothImageCom` DLL. The activity upon notification of activation creates an interface element, known as a `CameraPreview`, which is tied to the camera. The `CameraPreview` object provides a mechanism to display a continuous preview of the cameras output. In order to

allow for a Bluetooth connection, it also creates a thread which waits for a socket to be connected to by a client computer. This socket is configured to respond to Bluetooth communication with a particular Universal Unique Identifier (UUID) and service name chosen specifically this application. Upon receiving a connection from the client, an input and output stream are created to communicate with the client. The input stream serves to allow the client to make a command. It is via this input stream that camera configuration by the client could be implemented. After receiving the “take picture” command, the Bluetooth thread passes the output stream to a method which starts the process for taking a picture in Android and waits for notification that the following methods have completed writing the picture data. The wait and notification is implemented using the `wait` and `notify` methods of the `Java Object` class.

The image acquisition process consists of a series of event driven processes as detailed in Figure 4.6. A chain of callbacks are performed by the system to notify the application of the completion of each step. For this application, the callbacks are objects which implement a particular Java interface and therefore must include the necessary methods. First, the application must ask the camera to autofocus. Upon completing the focus, the system notifies the callback method provided in the previous step. This method then calls for the camera to take the picture providing two callbacks to notify when the picture has been taken and when the data is ready. The application is notified that the image data has been processed into JPEG format. To simplify the process of moving the data to the output stream, the output stream is simply passed through the chain of callback objects. This allows the callback method for handling the data processing complete event to write the JPEG image size and data directly to the output stream rather than passing it back to the Bluetooth thread. Finally, the Bluetooth thread is notified that the picture has been taken and written to the output stream and completes this process before preparing for a subsequent command.

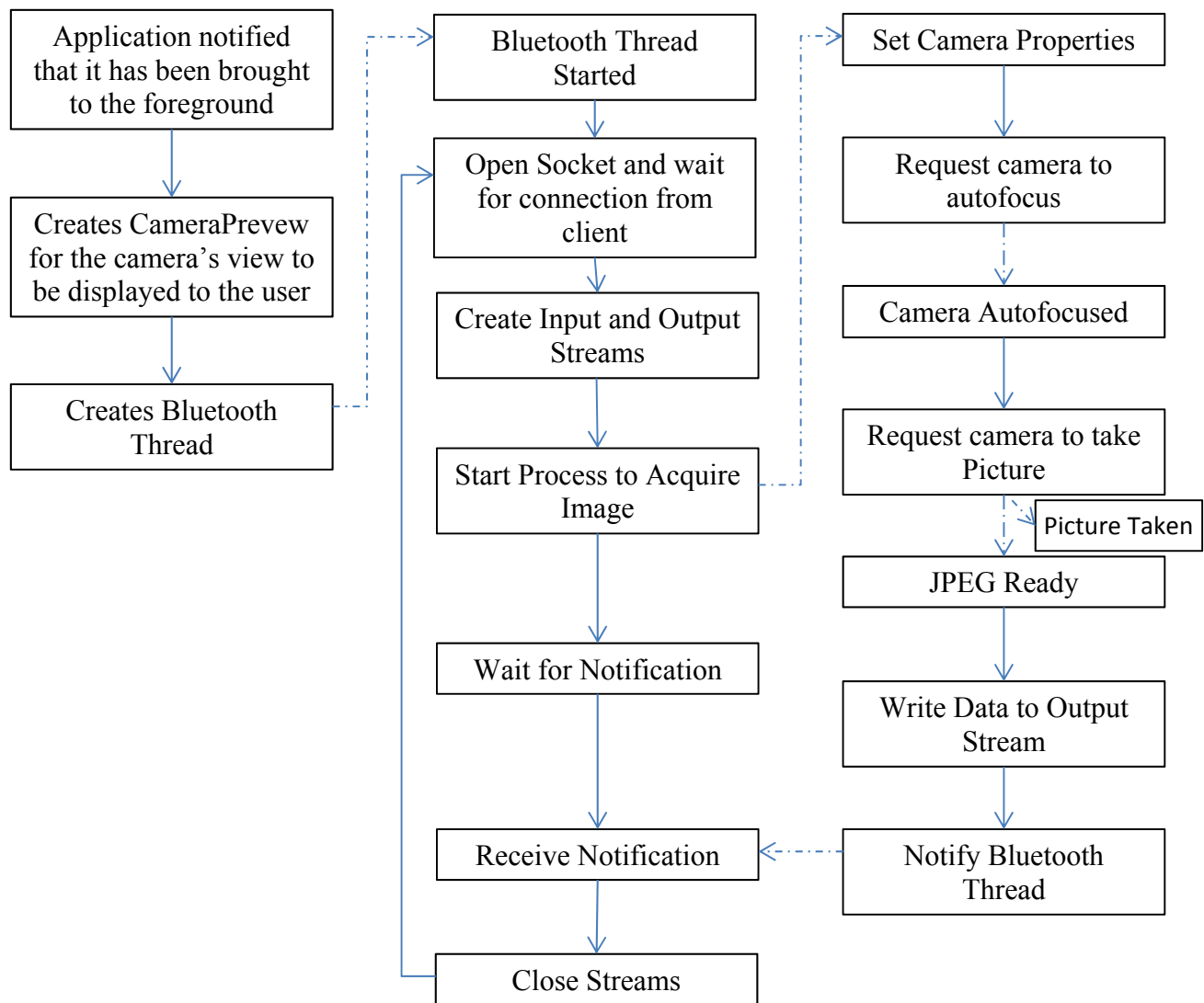


Figure 4.6 Bluetooth Scanner Android Process

4.3.13 Classroom Sync

Classroom Sync uses multicast network communication to synchronize the page viewed across multiple instances of VText on separate computers. First, the leader creates the session which VText assigns a random number as its identification number (ID). The session ID differentiates different Classroom Sync leaders from one another on a network. As part of the initialization process, the leader joins a multicast group. Every two seconds the leader sends the page they are viewing and their session ID to a multicast Internet protocol (IP) address and port

number using the Universal Datagram Protocol (UDP) as outlined in Figure 4.7. At the same time students set up as clients with the session ID provided by VText to the leader. They also join the multicast group, but then listen for the page and session ID. Upon receiving a message, they check if the session ID matches the session they are in. If the ID matches, they navigate to the page sent by the leader. Due to OneNote's storage scheme, page IDs do not match even if the book is copied from one computer to another. Because of this, VText utilizes the name of the page and the time the book was created. This does limit the utility of the application; however, it can be particularly useful when notebooks are shared between users. The name and date are used to look up the page in the page cache which finds the appropriate page ID. Classroom Sync then uses the OneNote API to navigate to the page matching the page on the lead computer. If the page is not found, the listener simply ignores the request. Both ends can temporarily pause leading and following to allow desynchronized navigation.

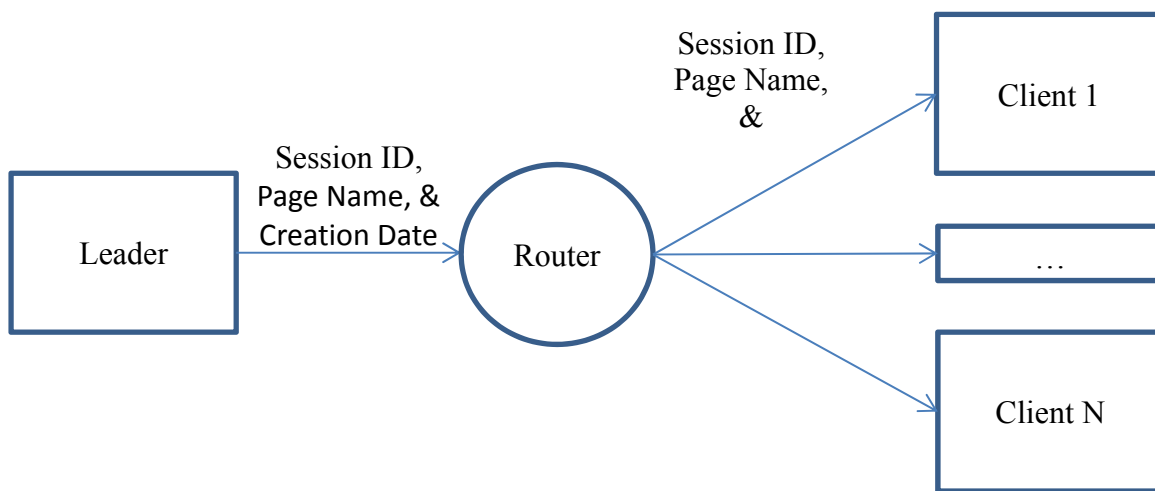


Figure 4.7 Multicast used to send page synchronization beacon

By using multicast, the lead computer is not required to manage a large number of sockets for each client instead allowing the infrastructure to handle the connections. Such a requirement could cause performance issues with a large number of users.

4.3.14 RTF Vector Graphics Rasterization

While OneNote supports inking, it does not support vector graphics such as shapes in Rich Text Format (RTF) documents. Therefore, before importing these images and drawings, they must be converted to a bitmap. This process is known as rasterizing. While there is no command in Word to simply convert drawings to bitmaps, when copied to the clipboard the images are saved in multiple formats including the original vector format, but also as a bitmap. To automate this process, a Word add-in is provided with VText which iterates through each drawing in a document open in Word). Each drawing is copied to the clipboard, deleted, and replaced with the bitmap stored on the clipboard. At this point the document can be saved as an RTF to work with the RTF Importer or saved as a PDF to be used with the PDF Importer.

4.3.15 RTF Importer (RTFON)

The RTFON importer utilizes the XDesigner RTF Document Object Model (DOM) Parser [41]. The parser interprets RTF documents notifying the program of images, tables, text, shapes, and page breaks in the document. The add-in utilizes this information to place tables, text, and images on the OneNote pages. Due to the lack of vector graphics support in OneNote, the process described in RTF Vector Graphics Rasterization is critical. Performing the rasterization without the extra Word add-in was explored, but ultimately did not produce satisfactory results. Unfortunately, this importer requires that the document be tagged with page breaks or split into separate files for each page to indicate where pages should be split. To simplify meeting this requirement, the Word add-in includes a method for splitting a document into multiple files. To achieve this, a copy of the original document is created and content is transferred page by page using the clipboard into a file for each page.

4.3.16 PDF Importer (PDFON)

The PDF importer utilizes the iText [42] library to parse PDF documents. The parsed content is then interpreted to produce OneNote page XML which can be inserted into pages in the OneNote notebook via the OneNote API. The interaction with iText and XML generation is performed by a Java program separate from the add-in. The results of the parsing operation are saved as an .onx file which is an XML compliant format consisting of a list of page elements

representing the eventual pages in OneNote. The resulting XML representing the content is inserted into the OneNote by the OneNote add-in.

The iText interpreter can handle text and images in PDF files. Due to the nature of PDF, the text is not guaranteed to be grouped in any meaningful way, rather each letter can simply be a separate element tagged with its location on the page. Therefore, paragraphs must be rebuilt from the individual text. To preserve page layout, each row is rebuilt and then placed into paragraphs based on the left coordinates of its bounding box. To fit into the OneNote text layout scheme, the line spacing for each row is estimated based on the distance between rows. Because documents have varying layouts, text sizes, kerning, and fonts, this required algorithmic regrouping of text provides varying levels of quality in the resulting OneNote content. Unfortunately, some documents may be imported unsatisfactorily in which case the PDFRON importer will be required. Images can then be placed using absolute coordinates into the empty space between these rows. To satisfy the OneNote API, the images are converted to Base64 text strings. PDF documents are not designed to allow text to reflow; therefore pagination is saved in the document. This simplifies VText's efforts to break the content into pages.

4.3.17 PDF Rasterized to OneNote (PDFRON) and PDF to Watermarked OneNote (PDFWON) Importers

While serving two separate purposes, the PDFRON and PDFWON Importers share a similar implementation. Both systems use the GhostscriptSharp library to generate images for each page [39, 45].

The PDFWON generates the .vti files through a separate .NET application which bundles each page into a .vti file. The .vti file is a zip archive of the images representing each page. First, the PDF is rasterized into a series of images without watermarks using GhostscriptSharp. The watermarking proceeds based on a text file of a list of strings to watermark. To create the watermarks, the image of the rasterized page created by GhostscriptSharp is stamped by the program to include the watermark string. The process of stamping text is repeated for each string in the text file creating a separate .vti file for each watermark string. The images are added to the .vti archive using DotNetZip. In contrast, the PDFRON importer runs directly in the add-

in storing the images in a temporary directory, but uses the same method for rasterizing pages through GhostscriptSharp.

The inclusion into OneNote is similar for both processes. For RTFWON, VText uses DotNetZip to extract each page image from the .vti archive. For RTFRON, VText reads from the temporary directory it created in the rasterization step. For each image, a new page is added using the OneNote API. The image is converted to Base64 format and inserted onto the page also using the OneNote API.

4.3.18 Text Importer

This importer is the simplest of the importers and runs directly from the add-in. It takes plain text files, which only contain characters, and imports them into OneNote. As the text file is read by VText, the number of characters is counted. After a character count threshold representing the number of characters which fit on half of a standard laptop screen at one time is met, the text import is continued on a new page at the next space or new line character. This keeps the content per OneNote page at a manageable size. By only switching to a new line at a space or new line character, words should not be split across multiple pages. The insertion of text and creation of new pages is handled through the OneNote API.

4.3.19 eQuiz System

The quiz taker is a separate .NET program bundled with VText. When installed, it assigns the eQuiz, .eqz. file format to be opened by the quiz taker. The eQuiz file is a simple XML file consisting of a top-level Quiz element and a series of Question elements. Only multiple choice questions are supported for now. The quiz taker builds a list of questions from the eQuiz file to be viewed one at a time. The possible answers for each question are specified in the eQuiz file and include feedback for each answer. Below in Code Snippet 4 is a sample three question quiz file. The `MultipleChoiceQuestion` element includes a `question` attribute which is displayed when taking the quiz. The `MultipleChoiceAnswer` children specify possible answers. Each answer includes an `answer` attribute which is displayed as solutions to the question specified by the `question` attribute when the quiz is taken. One answer per question has the `correct` attribute set to true. This allows the program to score the user's quiz. The

feedback attribute is displayed after the quiz to provide a hint or explanation of why that answer is right or wrong.

Code Snippet 4 Sample eQuiz File

```
<Quiz name="PSpice Electronics: Chapter 1">
  <MultipleChoiceQuestion question="What does SPICE stand for?">
    <MultipleChoiceAnswer correct="true" answer="Simulation Program with
Integrated Circuit Emphasis" feedback="Correct"/>
    <MultipleChoiceAnswer answer="Special Purpose Integrated Circuit
Engineering" feedback="Wrong"/>
    <MultipleChoiceAnswer answer="Simple Programming for Integrated Circuit
Engineering" feedback="Wrong"/>
    <MultipleChoiceAnswer answer="Simulation Protocol for Initial Circuit
Engineering" feedback="Wrong"/>
  </MultipleChoiceQuestion>
  <MultipleChoiceQuestion question="What are the 3 steps in designing a circuit?">
    <MultipleChoiceAnswer answer="Design, Build, Test" feedback="Wrong!"/>
    <MultipleChoiceAnswer correct="true" answer="Design Requirements,
Synthesize Circuit, Analyze Circuit" feedback="Correct!"/>
    <MultipleChoiceAnswer answer="Synthesize Circuit, Analyze Circuit, Design
Requirements" feedback="Wrong!"/>
    <MultipleChoiceAnswer answer="Requirements, Wait, Wait, Wait, Wait"
feedback="Wrong!"/>
  </MultipleChoiceQuestion>
  <MultipleChoiceQuestion question="How many transistors does the free version allow
you to use?">
    <MultipleChoiceAnswer answer="39" feedback="No, that is the number of analog
parts allowed."/>
    <MultipleChoiceAnswer correct="true" answer="10" feedback="In the paid
version you can use many more."/>
    <MultipleChoiceAnswer answer="11" feedback="Try one less!"/>
    <MultipleChoiceAnswer answer="9" feedback="Try one more!"/>
  </MultipleChoiceQuestion>
</Quiz>
```

4.4 Implementation Summary

This chapter has presented details on the various software platforms and libraries leveraged and components written to create VText. .NET, Java, Android, and OneNote provided functionality which allowed for development to focus upon features for electronic textbooks by reducing the amount of work required in building a foundation for VText. The following chapter provides analysis of the performance of the software components discussed in this chapter.

Chapter 5

Performance

Performance tests were run on two computers to compare the effects of certain design decisions on VText's usability. The traits tested include those that affect the programs reaction time to user input and power consumption. Reaction time and responsiveness are particularly important as it affects user satisfaction with software [46]. Power is important since users will often be using VText on laptops and tablets away from power outlets. The effects of creating the page cache, performing page look ups, running the system thread (see Section 4.3.2), and using the Bluetooth scanner on VText's usability are investigated.

The first of the two computers is a Dell Latitude E6400 running Windows 7 and Office 2010 with 4 gigabytes of RAM and an Intel Core 2 Duo P8700 clocked at 2.54 gigahertz. The hard drive performance was rated at 5.8 by the Windows System Assessment Tool. The second computer is a Fujitsu T4215 running Windows 8 and Office 2013 with 2 gigabytes of RAM and an Intel Core 2 Duo T5600 clocked at 1.83 gigahertz. The Fujitsu's hard drive was rated at 4.3 by the Windows System Assessment Tool. The Dell did not have a touch screen while touch is included on the Fujitsu. To measure times, the `C# Stopwatch` class from the `System.Diagnostics` library was utilized. `Stopwatch` allows timing of code execution down to ticks, which in the case of the Dell and Fujitsu computers, is approximately equal to 1000 processor cycles. Unless otherwise noted, millisecond resolution was used. To measure power consumption, Microsoft Research's Joulemeter was used on the Latitude [47]. Joulemeter estimates power consumption of various software operations based on processor cycle utilization.

5.1 Page Cache Lookup

The page cache is designed to reduce the amount of time it takes for the add-in to provide the page OneNote should navigate to based on a user's commands. Unfortunately, it cannot rely on searching the XML from the OneNote API to provide the information in a timely manner when there are a large number of pages open. While the cache lookup time also increases with the number of pages in the cache, even with nearly 12000 pages open, far more than a student would likely have in an electronic textbook, the lookup takes just over 1 millisecond. On the other hand, searching the XML takes 1.7 seconds with that number of pages. In a more reasonable environment of approximately 3000 pages, simulating four books of 500-1000 pages, the cache is still near 1 millisecond while searching the XML takes nearly 1 second. When combined with the time taken by OneNote to process the navigation request, it is important to reduce the cost and time of looking up the page to provide a fluid experience to the user.

5.2 Generating the Page Cache

Building and updating the page cache is not a quick process, but it allows VText to parse the XML representation of the pages provided by OneNote (see Section 4.1.6) only when necessary. Testing found that the time per page to build the cache was just over 1 millisecond. For 1000 pages this process would take just over 1 second. By saving the cache for future executions of the program, the startup time is reduced to about 10 milliseconds allowing the student access to the textbook content faster. The per page times were similar on the two test machines. Any differences can be attributed to the difference in speed and other factors like the number of other processes running on the machine. To determine when the cache must be rebuilt, a hash of the list of notebooks XML is calculated. This file changes only when a notebook is modified, such as by adding a new page. The hash is performed every iteration of the system thread. The hash function completes in less than 1 millisecond. If the hash has changed, the cache is rebuilt with the same timing conditions as the initial generation. This metric is important for considering power usage and providing updated data to properly perform tasks for the user.

5.3 System Thread

The system thread's timer and poll driven architecture (described in Section 4.3.2) requires a design that ensures it does not create a substantial burden on the machine since increased activity will increase power consumption [48]. Therefore, the first test measures the time taken to complete one iteration of the system thread. Both machines had similar results. The thread took on average 36 and 35 milliseconds on the Dell and the Fujitsu, respectively. Since the thread runs every 500 milliseconds, this represents 7% of total time available. The acceptability of this rate is analyzed in the next section.

5.4 Power Usage of VText

Joulemeter was utilized to test the power usage of VText. Pages were manually flipped using touch gestures at a rate of about one per second. Halfway through the test, two rebuilds of the page cache were induced by creating new pages. The data is represented in Figure 5.1. The two peaks are caused by the page cache being rebuilt. The duration of the plateaus match the duration from the previous testing described in Section 5.2 . It is important to also consider the power consumption of OneNote without the add-in since it is also required. Even with rapid page changes, OneNote rarely surpassed consuming 200 milliwatts on a system that idles at around eight Watts (one measurement found that the monitor is responsible for 3.1 Watts, idle hardware 4.4 Watts, and Windows .3 Watts). A similar test was performed while using Chrome to establish a comparison. This testing does not include network usage, only page rendering. Page changes and interactions with Chrome caused much larger spikes. While higher power usage by Chrome is expected, by demonstrating power usage lower than a common program, VText can be estimated to have power consumption levels acceptable to users.

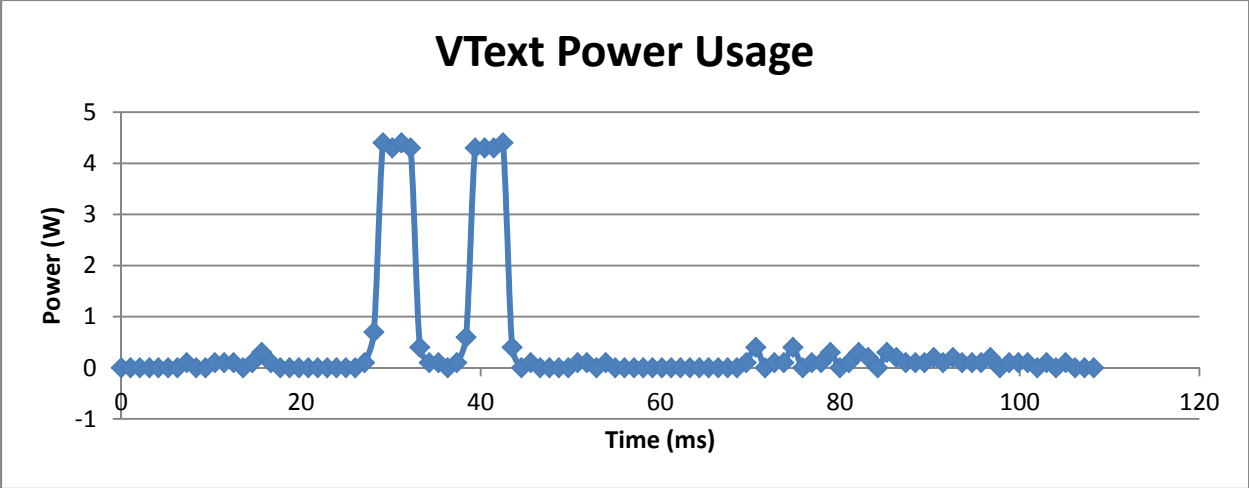


Figure 5.1 Power Usage of VText

5.5 Bluetooth Scanner Performance

The Bluetooth Scanner feature was also tested to determine the elapsed time from the user requesting to take a picture until the picture appears on the Windows machine’s screen. An HTC Evo 4G running Android version 2.3 was utilized as the photo capture device. The process of sending a picture request, taking the picture, and returning the image took a variable amount of time depending on the subject of the image. This is because the image is transferred as a Joint Photographics Experts Group (JPEG) file, a compressed image format. Therefore, the size of the image data changes as a function of the subject. To account for this, the test was performed on a blank sheet of black paper, a white document with black text and red ink, a whiteboard with a multitude of colors, and of an entire room. The results are shown in Table 1.

Table 1 Performance of Bluetooth Scanner

	Blank	Document	Whiteboard	Room
Average (sec.)	2.14	7.28	7.36	9.06
Standard Deviation	0.149	0.655	0.349	1.891

As expected, the whiteboard and document were similar, while the room took more time and the blank paper took less time. Shots of the entire room also experienced greater variance as the images were slightly different each time. Important for judging VText’s usability is that a student can have an image on his or her computer in 6 to 7 seconds. This is much faster than a

typical multifunction printer's scanning capability which can take upwards of 20 to 30 seconds as measured on an HP Officejet 4500. However, opening the application on the phone, lining up the camera, and then waiting this time may be too long. Therefore, a student likely would need to anticipate needing this functionality, possibly limiting its utility during lectures. One case when the application proves useful is to capture the results of meetings and discussions where colleagues have taken hand written notes on white boards and paper. While the quality of a normal scanner is higher, time and portability are critical for students attending lectures, participating in study groups, and working in groups on projects. The Bluetooth Scanner's performance appears to be acceptable in terms of file transfer time as well as providing reasonable image quality for cases where portability is required, but may require further work to reduce the setup and capture time.

5.6 Performance Summary

This chapter discussed the performance of VText in terms of response times and power usage. Two main performance requirements for electronic textbook usability are speed and power consumption. Speed is important as students are used to changing pages as rapidly as they can flip the page of a paper book. Power consumption is important because students are used to textbooks which do not require batteries. Sections 5.1 and 5.2 analyze the two most probable critical paths in providing readers with content. In both cases, software was implemented in the add-in to solve the issue of slow XML parsing by reducing the frequency it was required. This resulted in page navigation speeds limited by OneNote rather than the add-in by introducing a one-time initialization process the first time a book is loaded and a transparent process when pages are added, moved, or deleted in a book. Fortunately, OneNote itself rarely has any issues loading pages that are noticeable to the user. Section 5.5 discusses the limiting time factors of the Bluetooth Scanner process.

Sections 5.3 and 5.4 consider the power utilization of VText. This section found that VText's power requirements were less than that of Chrome providing a simplified comparison. The Dell Latitude the tests were run on is equipped with a battery rated for 56 Watt-hours. Based on the eight Watt measurement, this would suggest that the battery should last 7 hours although with WiFi, CD drive, and other programs and considering the true capacity of the

battery, this provides about three hours. The true meaning of the data is that VText has no special power or processing requirements over other software and therefore a potential user should consider the device's other intended activities when selecting the device, not VText.

Chapter 6

Usability Study and User Assessment

To measure the utility of VText as well as the user acceptance of all electronic textbooks, a survey was offered to 96 students enrolled in ECE 1574: Introduction to Problem Solving with C++ during the Spring 2013 semester. The majority of the students are second semester freshmen. Before being introduced to VText, a pre-survey was conducted to determine their opinions and background concerning electronic textbooks. Shortly after the pre-survey, they were offered portions of the course texts to use from within VText. Required Text 1 was offered as watermarked images (PDFWON) while the Supplementary Text 2 was distributed as a OneNote file which had been generated by the PDF to OneNote (PDFON) converter. Proof in the form of a receipt indicating that the hardcopy of Text 1 had indeed been purchased by all students who received the converted portion of the textbook was required. Text 2 was a supplementary textbook and is available online for free. Both books were used in class assignments, which forced the students to access the textbooks' material in order to successfully complete the assignment. At the end of the semester students were asked to complete a second survey concerning their impressions of VText.

6.1 Pre-Survey

Of the 96 students, 66 responded to the initial survey. Students were asked to report their study habits and classroom behaviors on a 5 point Likert scale, where one indicates that they strongly disagree and five indicates that they strongly agree. The data suggests that the use of an electronic textbook may increase the number of students who have their textbooks available in class. The mean score for "I brought a printed textbook to class" was 2.42 with 43% indicating

they never brought a textbook to class, while students overwhelmingly indicated that they brought a laptop to class with a mean of 4.49. Other data suggests students do not prefer to or actively collaborate with classmates. Data in Table 2 shows how students rated how often they collaborate. Interestingly, they indicated that they used the textbook to collaborate more than they shared their own notes.

Table 2 Data Relating to Student Collaboration

Question	Mean 1-Strongly Disagree 5-Strongly Agree
I used the textbook to collaborate with my classmates.	2.62
I communicated with the instructor by email.	3.06
I completed assignments on time.	4.88
I prepared for an exam.	4.43
I preferred to study alone.	3.66
I preferred to study in a group.	2.91
I procrastinated with reading and/or other assignments.	3.08
I talked to the instructor in person outside of class about ideas and concepts from the readings or classes.	1.89
I took notes in class.	4.02
I used class notes from others.	1.94
I shared my class notes with others.	2.26
I brought a printed textbook(s) to class.	2.42
I reflected on assigned readings before class.	2.38
I brought a laptop or similar device (e.g., netbook) to class.	4.49
I brought a tablet computer to class (iPad or similar).	4.08

The next section of questions asked students to report background information regarding their opinions on technology and electronic textbooks. As expected for students intending to major in electrical or computer engineering, all but 7% reported being generally comfortable

with computer-based technology, while 43% indicating they are very comfortable and typically seek out the latest hardware and software. Students were then asked if they agreed, using a 5 point Likert scale, with ways technology impacts education. Each of the statements suggests a positive influence. The results are included in Table 3. The students indicated that easy access to resources, organization, and communication with faculty were all impacted by technology.

Table 3 Student Opinion on Impact of Technology on Education

Question	Mean 1-Strongly Disagree 5-Strongly Agree
Technology facilitates access to resources.	4.66
Technology lessens administrative tasks for students.	3.87
Technology organizes student work.	4.42
Technology makes students feel more connected to other students.	3.95
Technology enhances communication from student to professor.	4.27
Technology enhances communication from professor to student.	4.48
Technology leads to increased student engagement in learning.	3.98
Technology improves student grades.	3.76

Students also gave background directly related to electronic textbook experience. They indicated that 29% had used electronic textbooks in high school and 69% had used electronic textbooks in college. They were then asked if they were apprehensive or eager to try textbooks using a 10 point Likert scale, where a one indicated apprehension and ten indicated eagerness. The mean rating was 6.9 indicating they were generally eager. A second question asked if they were comfortable with using an electronic textbook. The mean was 6.71, with a response of one for very uncomfortable and a ten for very comfortable, indicating they were generally

comfortable. This is similar to the earlier responses which indicated the students were comfortable with new technology.

Students were then asked about why they would or would not want to use electronic textbooks. While all of the suggested reasons, shown in Table 4, for electronic textbook use had favorable responses, those dealing with money and convenience were more important to students than pedagogical reasons [5, 4]. It is interesting that students indicated that sharing notes was important when earlier in the survey students indicated they did not frequently share their notes. The greatest concern of students, shown in Table 5, is that they will be easily distracted which has been listed as a concern and observed previously [3, 9]. However, students are generally not worried about receiving lower grades even with distractions. Interestingly, the preferences towards reading on screens or on paper, shown in Figure 6.1, are nearly even with a slight preference towards the former. This contrasts an earlier study of college level students performed in 2008 [15], but matches earlier studies of elementary students in 2005 [14]. While further study would be required to establish a true correlation, it is interesting to note that the students surveyed in this VText study were likely in 4th or 5th grade at the time of the earlier study. This would further support the theory that there is a generational shift in opinion towards electronic textbooks.

Table 4 Reasons for Using Electronic Textbooks

Question	Mean 1-Strongly Disagree 5-Strongly Agree
I can save money.	4.79
It is more convenient to download books.	4.31
I don't have to carry heavy books.	4.16
It will save time studying.	3.95
It will increase my academic performance (i.e., increase grades).	3.79
It will develop skills required for a good job.	3.69
I can share my notes.	3.68
I like to read on a computer.	3.42

It will help me graduate on time.	3.42
I will read more often.	3.29

Table 5 Reasons for Not Using Electronic Textbooks

Question	Mean 1-Strongly Disagree 5-Strongly Agree
I am easily distracted by other functions on my computer/e-reader (e-mail, Facebook, apps, etc.).	3.15
I prefer to read hardcopy books when I study.	3.13
I want to have access to the book after the course concludes.	3.13
I am afraid of losing material due to technology failure.	2.61
I will read less often.	2.11
The course instructor does not use it.	2
I will get lower grades.	1.89
I don't own a computer or tablet.	1.21

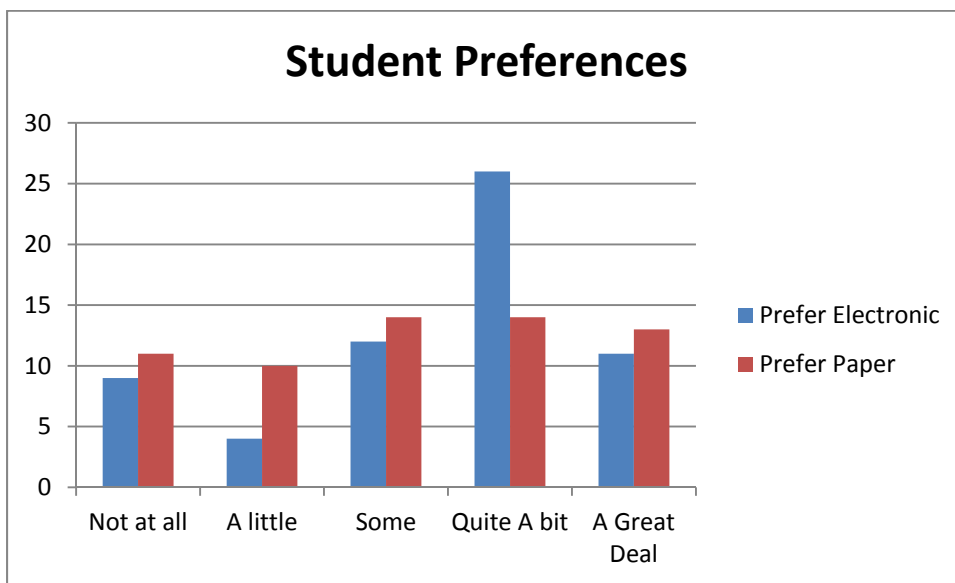


Figure 6.1 Student Preferences Towards Paper and Electronic Reading

Finally, students were asked how they foresaw the use of electronic textbooks impacting the course. Students did not believe they would be distracted or disengaged from the course nor were they worried about less efficient studying. There was some indication that they positively viewed the electronic textbook's ability to allow them to control how they were learning.

In conclusion, the pre-survey established a baseline for the students' opinions on electronic textbooks. It also contributed evidence to the theory that the change in preference of electronic textbooks is generational. These results, while interesting on their own, will be used to help draw conclusions on VText's utility and perhaps effectiveness.

6.2 Post-Survey

After having an opportunity to use VText as a framework for the electronic versions of Text 1 and Text 2, the students were asked to fill out a second survey to recount their experiences. This survey received responses from only 27 students. Students were first asked to judge if they read more, less, or the same during the trial than before the trial. Six students indicated they read more, while two indicated they read less. This outcome from using VText is better than the opposite, but is based on a sample size too small to draw strong conclusions. Four of the six students who said they read more also answered it was due to using VText, while one of the two students who said they read less answered it was due to using VText. Four students reported an increase in engagement with the course during the trial, while the rest reported no change. All four who reported an increase attributed the increase to using VText. While the response size is too small for producing strong conclusions, these results indicate that students perceive VText as capable of facilitating learning through better engagement and higher completion rates of assigned readings.

To assess the utility of VText displaying a single image representing each page (PDFWON), as in Text 1, versus maintaining text as text by using the PDF to OneNote (PDFON) converter, as in Text 2, students were asked to report their satisfaction with the two formats. To do so students were asked to respond using a seven point Likert scale ranging from very dissatisfied (one) to very satisfied (seven). The book generated using PDFWON was rated with a mean of 4.6, between neutral and somewhat satisfied, with a variance of 1.7. This compares unfavorably to the book generated using PDFON which was rated with a mean of 5.0 with variance of 1.23.

Students preferred the full text conversion with more students reporting being satisfied and fewer reporting being unsatisfied compared to the single image conversion. This is unsurprising as the full text version allows more flexibility for both display and interaction by allowing users to select and modify text for copying, reorganizing, editing and searching.

Students were asked to report how satisfied they were with the features of VText. The results are shown in Table 6 Student Satisfaction with VText Features. The students were asked if they used the various features and, if so, to rate their satisfaction with the particular feature. The results indicate that the most used features were searching the contents of a book, the navigational buttons, touch gestures, and side-by-side pages. Generally, the students were satisfied with the features they did use. In fact, the many neutral ratings may have been errantly recorded in place of “did not use”, which would lower the mean ratings towards four.

Table 6 Student Satisfaction with VText Features

Question	Did not use	Mean (Used VText) 1-Very Dissatisfied 7-Very Satisfied
Annotated/Highlighted text in the book	10	5.2
Bluetooth Scanner	18	4
Import Wizard	12	4.3
Navigational Buttons	8	4.9
Notes Page	13	5
Pinned Pages	15	4
Search the contents of the book	7	5.4
Search the Web	13	5.4
Side by side pages	9	5.3
Touch Gestures	9	4.6
User Guide	14	4.2

To target future development students were asked to identify which features they thought were most important even if they had not had a chance to try the feature out. The results are

shown in Table 7. Interestingly, students identified the Bluetooth Scanner, Import Wizard, Notes Page, and Search the Web as the least important features. It may be possible that as most students in this study had pen enabled computers, they did not need a method of importing notes not easily created with a mouse and keyboard. The Notes Page and Search the Web were likely identified as students can perform these tasks manually without VText. The importance of a general search the web feature would likely be rated higher if the question had been asked in the context of the device rather than VText. The Import Wizard may have been ranked low as they assumed importing other formats into OneNote would not be their task, but instead be completed by publishers or instructors. However, it is worth noting that while students identified these features as less important than others, they were not identified as unimportant.

The most important features identified by students were Searching the Contents of the Book, Navigational Buttons, Pinned Pages, Side-by-Side Pages, Annotations and Highlighting, and Touch Gestures. These selections are interesting as they, except Searching the Contents of the Book, are features which bring the electronic studying experience in line with that of reading in a paper textbook. However, Searching the Contents of the Book can be considered to be an improvement on paper textbook indexes. These answers show that students do not want to sacrifice any part of paper textbooks, but have different views of which new features are important.

Table 7 Importance of Features in VText

Question	Mean 1-Very Unimportant 7-Very Important
Annotated/Highlighted text in the book	5.33
Bluetooth Scanner (ability to immediately take a picture with your phone and have it on your Windows Laptop)	4.26
Import Wizard (Import PDF's, RTF's and text files into OneNote)	4.85
Navigational Buttons (ability to flip to the previous and next pages with one small button)	5.63
Notes Page (Copy content from the book onto a notes	5.11

page)	
Pinned Pages (Bookmark pages in the book for later review)	5.37
Search the contents of the book	6.35
Search the Web (Search the web for keywords selected in the text)	5
Side by side pages (Can see two pages on the screen at one time)	5.37
Touch Gestures (ability to change pages or pin pages with quick gestures)	5.33

While earlier studies lead to lower desire to use electronic textbooks in the future [15], the survey indicates that students were either not influenced by the trial or influenced positively. Of 27 responses, 14 indicated no change while 13 indicated being more interested in trying electronic textbooks in the future. Table 8 shows the level of use and the change in interest. Students who answered “I used just the provided OneNote file, but did not install VText” used the provided OneNote version of the supplementary textbook, but did not install the VText software. Interestingly, 47% of students who did not use VText still reported an increase in interest, while 53% of students who used VText reported an increase in interest. This effect may be explained by efforts during the class by the instructors to introduce students to electronic textbooks. Further, students who were motivated to try VText are more likely to have been interested in the first place.

Table 8 Cross-tabulation of VText Use and Interest in Using Electronic Textbooks in the Future

	What was the primary way you read the e-Textbook?			
How have your feelings toward eTextbook use changed since before the trial period? Are you more or less interested in using them in the future?	I used VText	I used just the provided OneNote file, but did not install VText	I didn't use VText or OneNote	Total

Much Less	0	0	0	0
Less	0	0	0	0
Somewhat Less	0	0	0	0
The Same	6	8	0	14
Somewhat More	3	2	2	7
More	4	1	1	6
Much More	0	1	0	1
Total	13	12	3	28

6.3 Assessment Conclusions

This study provides feedback on the design of VText. The results indicate that VText had a positive influence on students' opinions of electronic textbooks, even if VText may not be their preferred platform. Students provided further evidence for Marmarelli and Young's [5, 4] conclusions that money and convenience are more important factors than pedagogy. Students also provided interesting results concerning which features they value in an electronic textbook. For instance, they responded that searching the web from the textbook is less important than searching the content of a textbook. Finally, students also showed an increase in interest in electronic textbooks from before the study and from the results of previous studies. This result suggests that further research and product development with VText and electronic textbooks in general would be welcomed by students.

Chapter 7

Conclusions

Unlike other industries, or even other sectors of publishing, textbook publishing has not successfully shifted towards electronic platforms. This is the result of devices not meeting the expectations of nor generating excitement amongst students and faculty. Proposed solutions often fall short for a combination of poor design, poor pricing, and even student resistance to changing their study methods. However, devices which are easy to use and are not unfamiliar to students have a chance to generate interest if costs are acceptable to students. Faculty and schools are more likely to encourage adoption if solutions provide active reading opportunities which have been shown to increase academic performance.

VText supports active reading by allowing notes from reading to be in the same location as the textbook content and lecture notes. This helps create context between information sources, an important component of active reading. Active reading and class engagement is further enhanced by allowing students to search the web directly from the book and by collocating the textbook on a full powered laptop or tablet.

The features provided by VText should improve student performance; however, student performance was not measured by the survey conducted during the spring semester of 2013. Therefore, performance should be measured by further study including a semester long test of VText with a fully imported and edited textbook. While more assessment is needed, the results of the 2013 survey are promising in that students generally reported the experience to be positive. This indicates that students in the future may also be willing to try an electronic textbook.

References

- [1] T. Davy, "E-Textbooks: Opportunities, innovations, distractions and dilemmas," *Serials: The Journal for the Serials Community*, pp. 98-102, 2007.
- [2] J. R. Miller, A. W. Nutting and L. Backer-Eveleth, "The Determinants of Electronic Textbook Use Among College Students," *Cornell Higher Education Research Institute*, 2012.
- [3] D. B. Daniel, "E-Textbooks at What Cost? Performance and Use of Electronic v. Print Texts," *Computers & Education*, vol. 62, pp. 18-23, 2012.
- [4] J. R. Young, "6 Lessons One Campus Learned about E-Textbooks," *Chronicle of Higher Education*, vol. 55, no. 39, 2009.
- [5] T. Marmarelli and M. Ringle, "The Reed College Kindle Study," 2009.
- [6] L. A. Clyde, "Electronic Books," *Teacher Librarian*, vol. 32, no. 5, p. 45, 2005.
- [7] J. Hattie, J. Biggs and N. Purdie, "Effects of Learning Skills Interventions on Student Learning: A Meta-Analysis," *Review of Educational Research*, vol. 66, no. 2, pp. 99-136, 1996.
- [8] J. Tront, "Operations Needed in e-Textbooks of the Future," Blacksburg, VA, 2010.
- [9] T. Marmarelli and M. Ringle, "The Reed College iPad Study," 2011.
- [10] A. Martinez, "Amazon.com's Kindle Fails First College Test," *Seattle Times*, 2010 5 2010.
- [11] A. J. Rockinson-Szapkiw, J. Courduff, K. Carter and D. Bennett, "Electronic Versus Traditional Print Textbooks: A Comparison Study on the Influence of University Students' Learning," *Computers & Education*, vol. 63, pp. 259-266, 2013.
- [12] D. Reinking, "Electronic Literacy. Perspectives in Reading Research No. 4," 1994.
- [13] K. Hornbæk and E. Frøkjær, "Reading of Electronic Documents: The Usability of Linear, Fisheye, and Overview+ Detail Interfaces," in *Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems, 2001.

- [14] S. Maynard and E. Cheyne, "Can Electronic Textbooks Help Children to Learn?," *The Electronic Library*, vol. 23, no. 1, pp. 103-115, 2005.
- [15] J. A. Shepperd, J. L. Grace and E. J. Koch, "Evaluating the Electronic Textbook: Is it Time to Dispense with the Paper Text?," *Teaching of Psychology*, vol. 35, no. 1, pp. 2-5, 2008.
- [16] R. Aust, M. J. Kelley and W. Roby, ""The Use of Hyper-References and Conventional Dictionaries," *Educational Technology Research and Development*, vol. 41, no. 4, pp. 63-74, 1993.
- [17] K. A. Kiewra and S. L. Benton, "The Relationship Between Information-Processing Ability and Notetaking," *Contemporary Educational Psychology*, vol. 13, no. 1, pp. 33-44, 1988.
- [18] K. Kim, S. A. Turner and M. A. Pérez-Quiñones, "Requirements for Electronic Note Taking Systems: A Field Study of Note Taking in University Classrooms," *Education and Information Technologies*, vol. 14, no. 3, pp. 255-283, 2009.
- [19] "HP WebOS," [Online]. Available: www.hpwebos.com/us. [Accessed 24 11 2013].
- [20] B. N. Schilit, G. Golovchinsky and M. N. Price, "Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations," in *Proceedings of The SIGCHI Conference on Human Factors in Computing Systems*, 1998.
- [21] M. J. Adler and C. van Doren, *How to Read a Book*, New York, NY: Simon and Schuster, 1972.
- [22] N. Yankelovich, N. Meyrowitz and A. van Dam, "Reading and Writing the Electronic Book," *Computer*, vol. 18, no. 10, pp. 15-30, 1985.
- [23] A. Adler, A. Gujar, B. L. Harrison, K. O'Hara and A. Sellen, "A Diary Study of Work-Related Reading: Design Implications for Digital Reading Devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1998.
- [24] L. W. Foderaro, "In a Digital Age, Students Still Cling to Paper Textbooks," *The New York Times*, p. A21, 19 10 2010.
- [25] "XLibris," [Online]. Available: <http://www.fxpal.com/?p=xlibris>. [Accessed 1 3 2013].
- [26] C. Dawson, "Updated: Will the Kno Tablet Finally make e-Textbooks Mainstream?," 27 9 2010. [Online]. Available: <http://www.zdnet.com/blog/education/updated-will-the-kno-tablet-finally-make-e-textbooks-mainstream/4251>. [Accessed 1 3 2013].
- [27] "Kno," [Online]. Available: <http://www.kno.com>.

- [28] C. Dawson, "The Kno is Dead...Long Live Kno," 11 4 2011. [Online]. Available: <http://www.zdnet.com/blog/education/the-kno-is-dead-long-live-kno-thanks-to-30-million-from-intel-et-al/4545>. [Accessed 1 3 2013].
- [29] "CourseSmart," [Online]. Available: <http://www.coursesmart.com/>. [Accessed 17 4 2013].
- [30] "VitalSource," [Online]. Available: <http://www.vitalsource.com/>. [Accessed 17 4 2013].
- [31] "Chegg," [Online]. Available: <http://www.chegg.com>. [Accessed 17 4 2013].
- [32] "Flatworld Knowledge," [Online]. Available: <http://www.flatworldknowledge.com>. [Accessed 17 4 2013].
- [33] "Microsoft .NET," [Online]. Available: <http://www.microsoft.com/net>. [Accessed 17 4 2013].
- [34] Microsoft, "Introduction to the C# Language and the .NET Framework," [Online]. Available: <http://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>. [Accessed 17 4 2013].
- [35] "Java," [Online]. Available: <http://www.java.com/en>. [Accessed 17 4 2013].
- [36] Microsoft, "Windows Presentation Foundation," [Online]. Available: <http://msdn.microsoft.com/en-us/library/ms754130.aspx>. [Accessed 2013 4 2013].
- [37] Microsoft, "Welcome to the OneNote Reference," [Online]. Available: [http://msdn.microsoft.com/en-us/library/office/ff925947\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/office/ff925947(v=office.14).aspx). [Accessed 17 4 2013].
- [38] "Android Development," [Online]. Available: developer.android.com. [Accessed 17 4 2013].
- [39] "Ghostscript," [Online]. Available: <http://www.ghostscript.com>. [Accessed 17 4 2013].
- [40] M. Ephraim, "Introducing GhostscriptSharp," 8 Jul 2008. [Online]. Available: <http://mattephraim.com/blog/2009/07/08/introducing-ghostscriptsharp/>. [Accessed 26 4 2013].
- [41] "RTF DOM Parser," [Online]. Available: <http://rdp.codeplex.com>. [Accessed 17 4 2013].
- [42] "iText," [Online]. Available: <http://itextpdf.com>. [Accessed 17 4 2013].
- [43] "DotNetZip - Zip and Unzip in C#, VB, any .NET Language," [Online]. Available: <http://dotnetzip.codeplex.com>. [Accessed 17 4 2013].
- [44] "32feet.NET - Personal Area Networking for .NET," [Online]. Available: <http://32feet.codeplex.com>. [Accessed 17 4 2013].

- [45] "PDFsharp," [Online]. Available: www.pdfsharp.com. [Accessed 17 4 2013].
- [46] E. G. Nilsson, "Design Patterns for User Interface for Mobile Applications," *Advances in Engineering Software*, vol. 40, no. 12, pp. 1318-1328, 2009.
- [47] Microsoft Research, "Jolueometer," [Online]. Available: research.microsoft.com/jump/131778. [Accessed 17 4 2013].
- [48] A. Mahesri and V. Vardhan, "Power Consumption Breakdown on a Modern Laptop," in *Power-Aware Computer Systems*, Portland, OR, 2004.
- [49] S. Paxhia, "The Challenges of Higher Education Digital Publishing," *Publishing Research Quarterly*, vol. 27, no. 4, pp. 321-326, 2011.

Appendix A User Guide

A.1 VText

This document provides information for using and troubleshooting VText. Installation information can be found in Installation/Uninstallation/Upgrade and troubleshooting information can be found in Troubleshooting. This document will be updated as issues arise during initial testing.

VText adds features to Microsoft OneNote to allow it to work as a textbook viewer while preserving OneNote’s strengths in note taking. VText allows students to easily utilize Windows based tablets, slates, and PCs for efficient reading. It is our belief that the seamless navigation provided by VText and limitless annotation strengths provided by OneNote will encourage and simplify active reading. VText is a research project from Virginia Tech by John Cristy and Joseph Tront.

A.2 System Requirements

VText still needs to be tested on a wide range of devices. Below “Full” means well-tested and “Limited” means limited testing and certain features may not work.

	Full	Limited
Operating System	Windows 7	Windows 8 (VText gestures may not work)
Microsoft Office*	2010	2013 (VText gestures may not work)
Processor	Core 2, i3, and better	Atom-based
Storage	The method of importation, content, and size of book can change this value drastically. If importing a PDF as images, the size can reach 500 MB, otherwise it will be 20-50 MB. Multimedia will increase the book by the size	

	of the file inserted into the text. VText's installation is < 50 MB
Pen/Touch	Preferred, but not required
Bluetooth and Android Device	Required only for the Bluetooth scanner

*VText runs with full OneNote, not OneNote MX

A.3 Installation/Uninstallation/Upgrade

To Install VText, OneNote must be closed. Then run VText's installation .msi file. To Uninstall VText simply locate eTextbook in Programs and Features, then click Uninstall. Unfortunately, upgrading VText requires uninstalling the old version before installing the new version. Simply follow the uninstall instructions followed by the install instructions above.

A.3.a Install

1. Execute the provided vText<version>.msi file.
2. The following window should open. Select "Next".

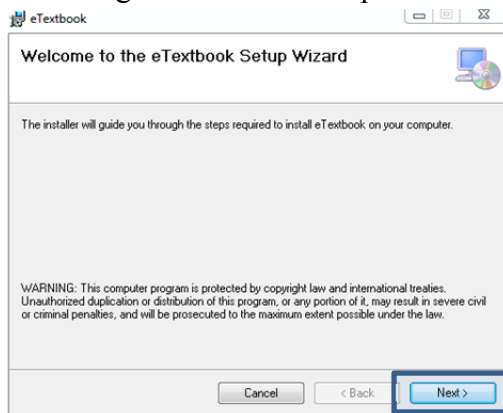


Figure A.1 Installation Wizard (Used with permission from Microsoft.)

3. You may now change the installation path as well as who has access to the program, then click "Next"

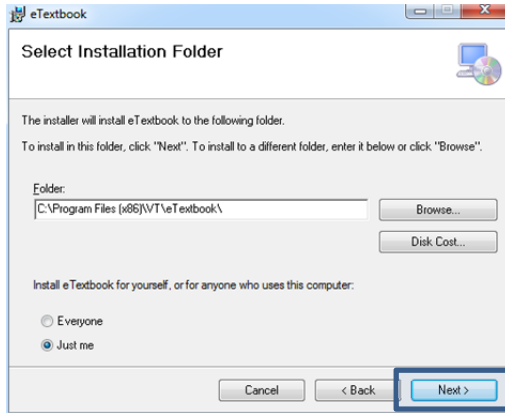


Figure A.2 Installation Options (Used with permission from Microsoft.)

4. Click “Next” again and wait for the add in to complete installation. If prompted to grant permission to complete the installation, click “Yes”.
5. Launch OneNote. You may see the following window pop up if this is the first time VText has been installed on your computer. Allow this to complete, it should not take long unless you have many OneNote Notebooks.

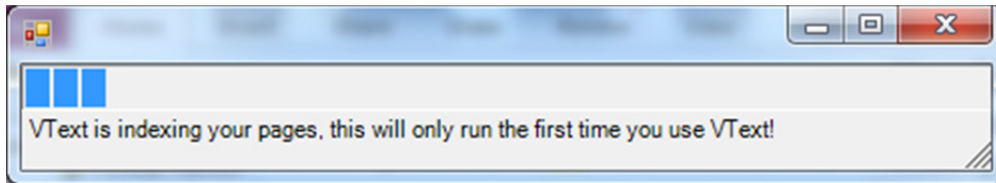


Figure A.3 Index pages progress window (Photo by author, 2013)

6. If the add in does not load automatically the first time you open OneNote after installation, you should follow these steps.
7. In OneNote, click “File” then “Options”.

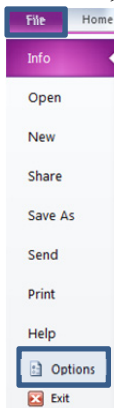


Figure A.4 OneNote Menu (Used with permission from Microsoft.)

8. Then click “Add-Ins” followed by “Go...”

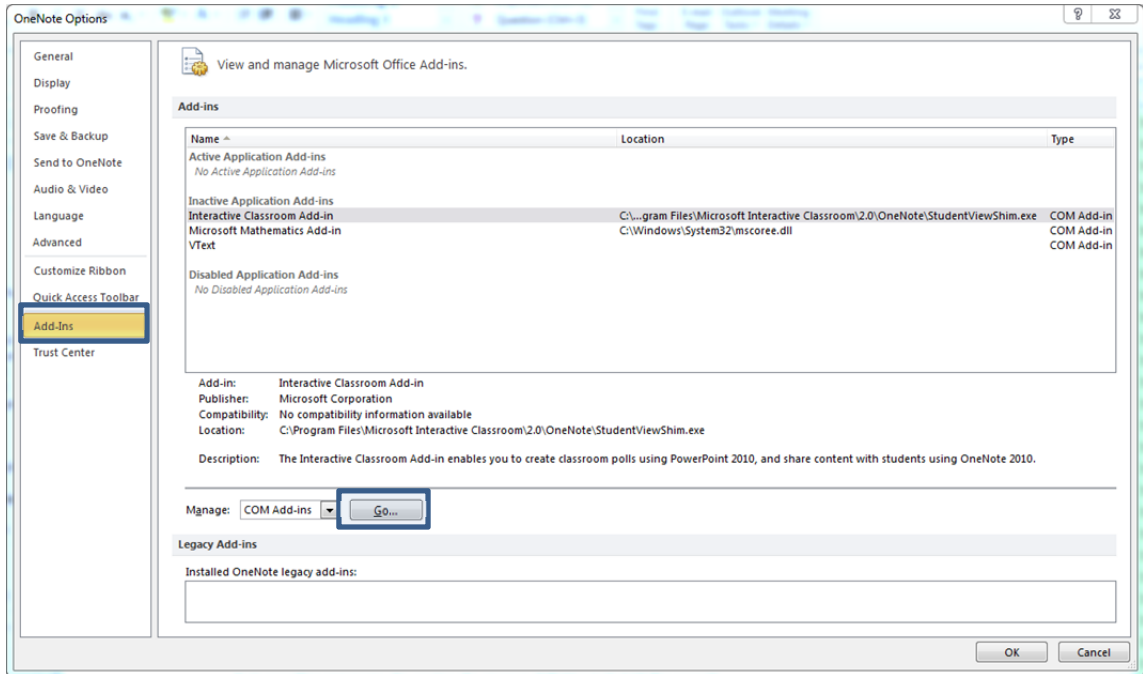


Figure A.5 OneNote Menu (Used with permission from Microsoft [37])

9. Check the box labeled “VText”, then click “OK”.

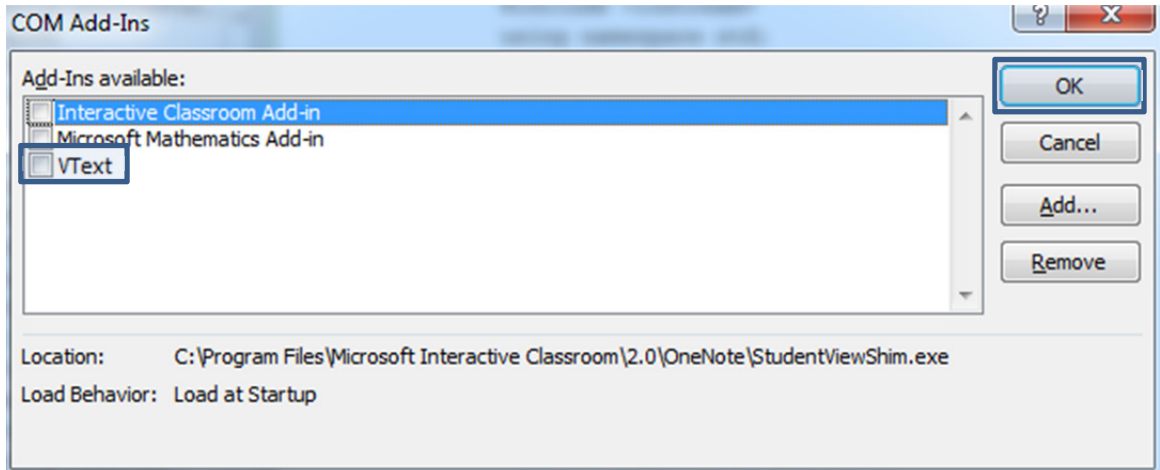


Figure A.6 OneNote Add-Ins menu (Used with permission from Microsoft [37])

A.3.b Uninstall

1. Open the Windows’ Control Panel and select “Uninstall a Program” or “Programs and Features” depending on the “View By” selection.

2. Find eTextbook and click Uninstall.

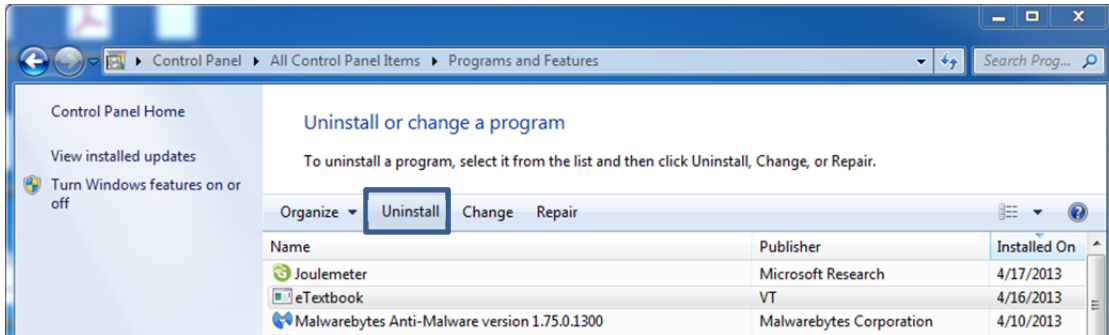


Figure A.7 Uninstallation Procedure (Used with permission from Microsoft [37])

3. If you are asked if you are sure, click “Yes”.
4. VText should be uninstalled.

A.3.c Upgrade

Follow the Uninstall instructions followed by the Install instructions.

A.4 VText Controls

VText adds controls to the program in four ways: 1) New tab on the Ribbon, 2) on-page controls, and 3) new gestures to allow for easier navigation 4) additional context menu options on right click.

A.4.a Overview of New Tab “VText”

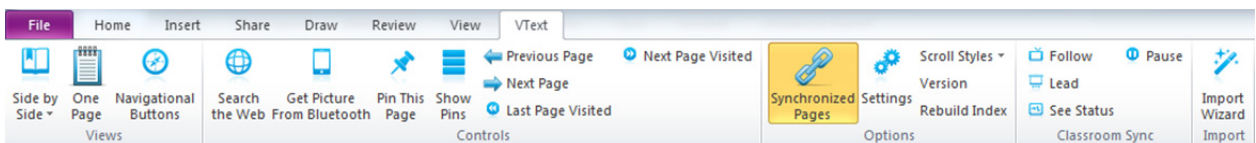


Figure A.8 Ribbon tab added to OneNote by VText (Used with permission from Microsoft [37])

Button	Action
Side by Side	Opens two windows side by side (<i>See Controlling the Windows</i>)
Side-by-Side -> Vertically Tiled	Opens two windows top to bottom (<i>See Controlling the Windows</i>)
Side-by-Side -> Homework	Opens two windows to the same page to allow for editing the page in different areas at the same time(<i>See Controlling the Windows</i>)

Mode	
One Page	Opens one page in full screen mode
Navigational Buttons	Adds navigational buttons to the window to allow one button page changes (<i>See Basic Navigation</i>)
Search The Web	Allows the user to search the web for more information on selected content (<i>See Search the Web</i>)
Get Picture from Bluetooth	Opens the dialog for using the Bluetooth Scanner (<i>See Bluetooth Scanner</i>)
Pin This Page	Pins the open page for easy access (<i>See Basic Navigation</i>)
Show Pins	Displays pinned pages as well as a list of all notebooks (<i>See Basic Navigation</i>)
Previous Page	Navigates to the previous page
Next Page	Navigates forward one page
Last Page Visited	Allows you to jump back to the page you were previously on
Next Page Visited	Allows you to undo the “Last Page Visited” command
Synchronized Pages	If two windows are open, the windows will always show consecutive pages (<i>See Controlling the Windows</i>)
Settings	Allows the user to adjust certain VText Settings
Scroll Styles	Allows the user to select two options, book style and scroll style. In book style, two windows navigating forward will change for example from page 1 and 2 to 3 and 4 like a book. In scroll style, two windows navigating forward will change from page 1 and 2 to 2 and 3, like a scroll.
Version	Shows the current version of VText installed
Rebuild Index	Possible step in fixing VText (<i>See Troubleshooting</i>)
Follow	Computer will sync to page navigation on the leading device (<i>See Classroom Sync</i>)
Lead	Computer will sync other computer to page navigation on this device (<i>See Classroom Sync</i>)
See Status	Shows information about the current Classroom Sync session (<i>See Classroom Sync</i>)
Pause	Pauses and restarts a session. If following, navigation on the lead device will not show on this machine. If leading, navigation on the lead device will not be broadcast to following machines. (<i>See Classroom Sync</i>)
Import Wizard	Opens the import wizard (<i>See Importing Content</i>)

A.4.b Overview of Settings

Option	Description
Search	This changes the search engine that will be used for the Search the Web

Engine	feature. Currently Bing, Google, Yahoo, Wikipedia, and Reddit’s AskScience are available.
Touch Gestures	This allows the user to disable the touch gestures described in Basic Navigation.
Pen Gestures	<i>Unavailable</i>
Mouse Gestures	<i>Unavailable</i>
Open add-ins right click menu to the left	Changes how the right click menu opens. It should be left at the default unless described in the troubleshooting.
Sort Pinned Pages by Notebook	Controls how the pinned page listing are organized (<i>See Pins Page</i>)

A.5 Controlling the Windows

VText allows the user to quickly and easily open two windows side-by-side or top-to-bottom. These windows can be synchronized by clicking on “Synchronized Pages.” When synchronized, the pages will both change much like how flipping a page in a book changes both views. The two views can also be desynchronized to allow the student to read from different books or different notebooks at the same time. There is also a “Homework” mode which allows a user to open the same page side-by-side to allow the student to work in one area of a page while viewing the problem in another area.

A.6 Basic Navigation

While the author of a textbook can manipulate the pages of a VText textbook, generally a textbook will be a section within a notebook. The standard OneNote interface can be used to navigate books, but VText adds methods for navigating while preserving screen space in full screen mode. There are two options gestures and buttons.

VText adds gestures to OneNote’s pan and pinch to zoom gestures. These gestures allow for “Pinning,” or bookmarking, and moving forward and backward through the book. The same can be done on non-touch computers by utilizing the navigation buttons shown below which can be brought up by clicking “Navigation Buttons” on the VText tab of the ribbon. Clicking and

dragging up on the navigation buttons shows the option to open the pinned pages view and pin the current page. The pin and show pins functions can also be accessed on the VText tab as well as by clicking and dragging up on the navigation buttons.

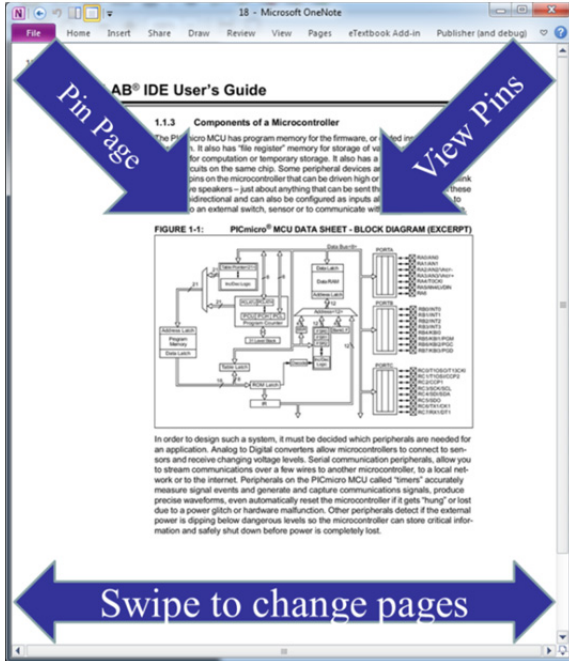


Figure A.9 VText Gestures (Used with permission from Microsoft [37])

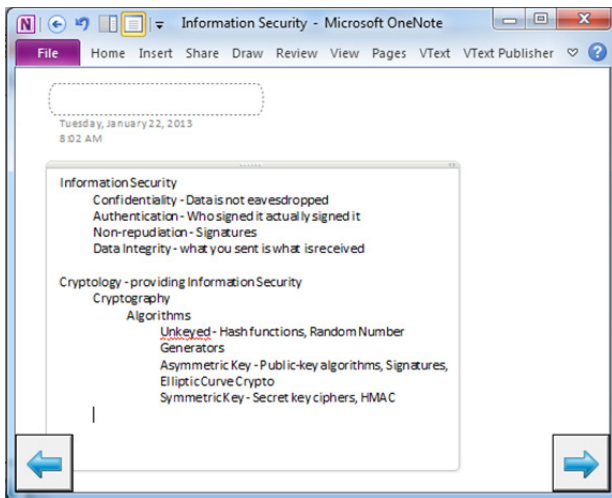


Figure A.10 Navigation Buttons (Used with permission from Microsoft [37])

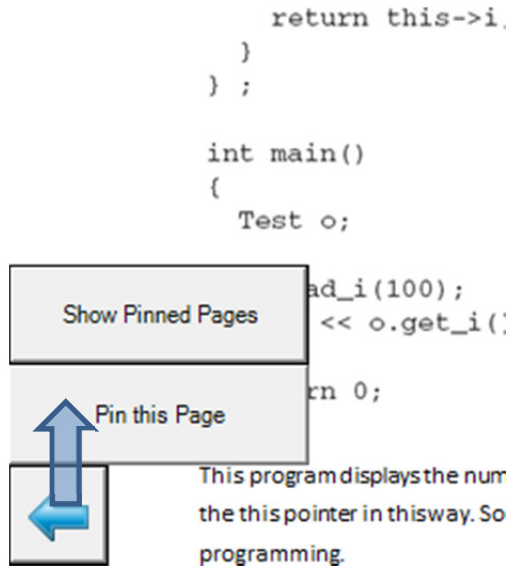


Figure A.11 Navigation Buttons Menu (Photo by author, 2013)

A.7 Pins Page

The pins page which opens for the “Show/View Pins” button and gesture provides links back to pinned pages. It also provides a second tab listing all open notebooks for further navigation. Pinned pages can be removed by right-clicking on the button for the particular page. The pin listings can be sorted based on when the pages were pinned or based on the notebook they are in. The pinned pages listing also scrolls automatically to the first pin in the current notebook.

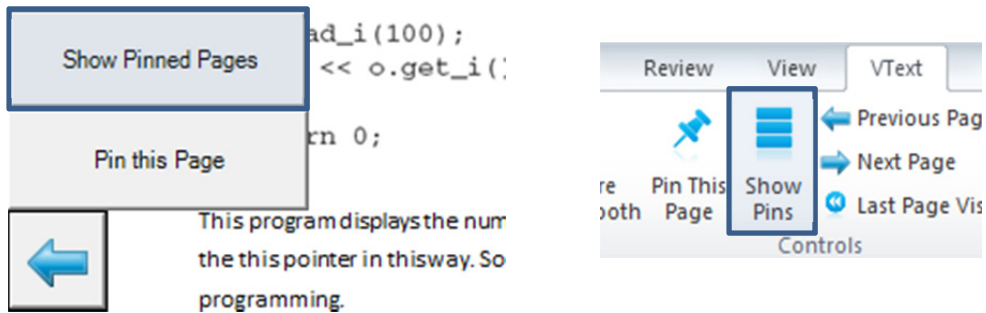


Figure A.12 Methods of opening the pinned pages window (Photo by author, 2013)

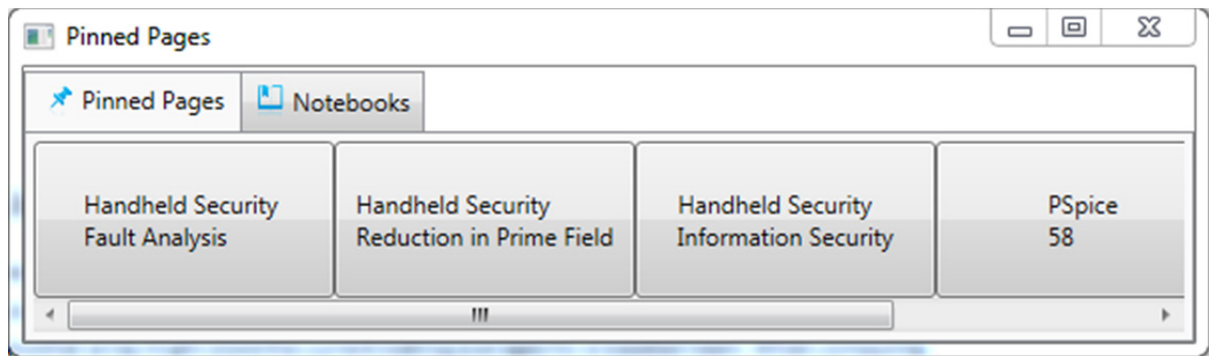


Figure A.13 The pinned pages window (Photo by author, 2013)

A.8 Search the Web

VText allows readers to search the web for terms found in the textbook. Unfortunately, for now the user can only search text and not handwritten notes. To search, the user should select the text, right-click (touch and hold with a pen), then click on the Search Bing button that pops up. The default web browser should open with the selected text searched. The search engine, Bing is the default, can be chosen on the Settings menu of VText. It can also be accessed by clicking on the Search the Web button on the ribbon.

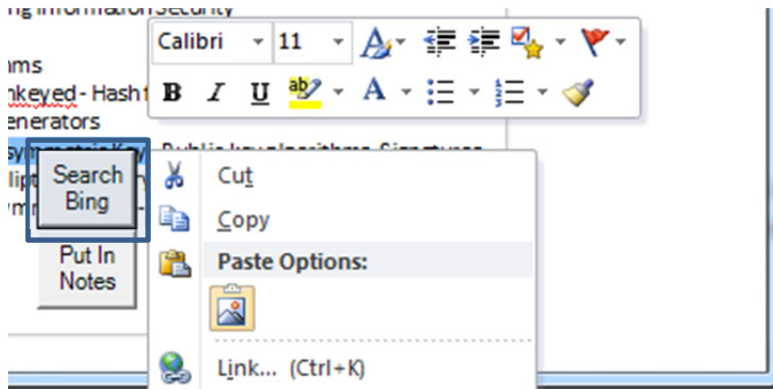


Figure A.14 Search the web from the right click menu (Used with permission from Microsoft [37])

A.9 Put In Notes

VText allows students to have notes copied automatically to a centralized notes page. The “Put In Notes” button is accessed by selecting text, then right clicking, much like the search the web function.

In order for this to work, there must be a page in the front of the section called “Notes”. Generally, this should be added by the author or instructor before distribution. VText will place the content in the “Notes” page preceding the source page.

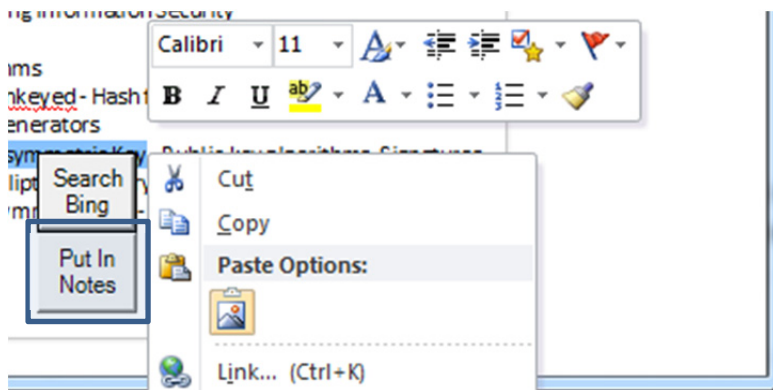


Figure A.15 Copy text to notes from the right click menu (Used with permission from Microsoft [37])

A.10 Classroom Sync

Classroom Sync allows teachers and students to be on the same page automatically to save time during lectures. Classroom Sync does not share content, just which page is being viewed. Therefore, the teacher and students must have the same book. A teacher should click “Lead” on

the ribbon. The program will then give a session ID. This number should be given to students to enter after clicking “Follow.” Both the students and the teacher can pause participating by clicking “Pause.” The session is automatically ended when OneNote is shut down.

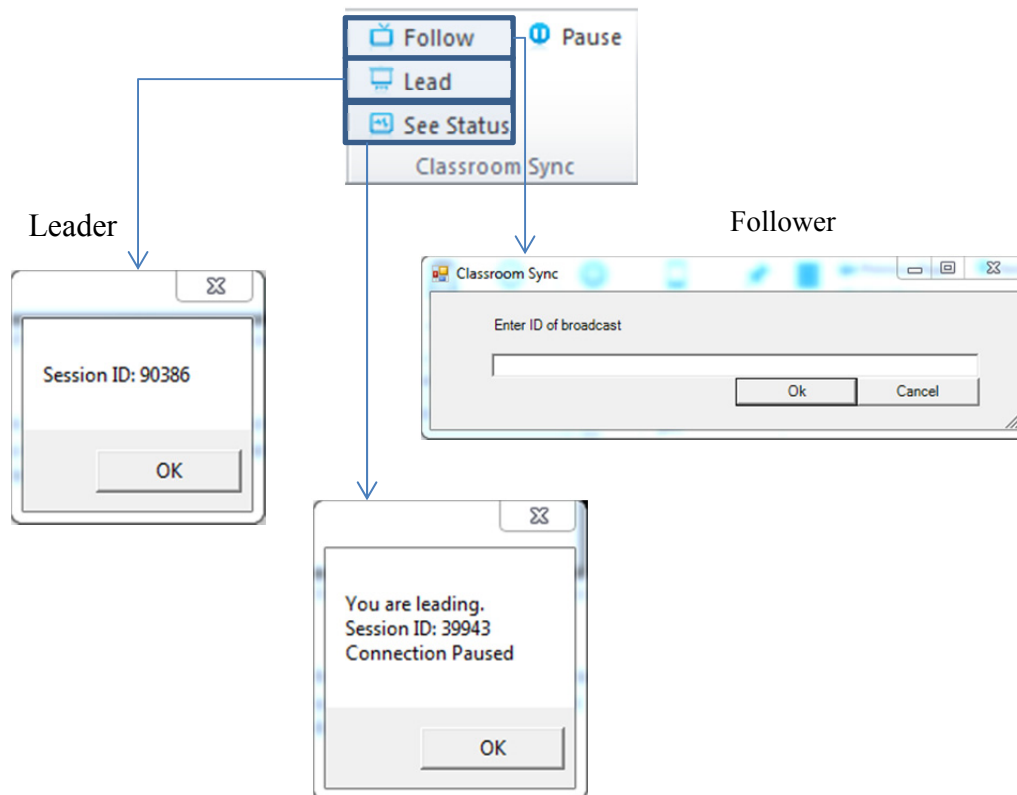


Figure A.16 Elements of the Classroom Sync feature (Photo by author, 2013)

A.11 Bluetooth Scanner

The Bluetooth Scanner allows an Android phone to immediately send a photo to your computer seamlessly. You may have to pair your phone and computer (see your phone’s documentation, however, it usually simply involves making the phone’s Bluetooth visible and clicking “Add A Device” in Windows). To use the Bluetooth Scanner:

1. Click “Get Picture From Bluetooth” in OneNote/VText

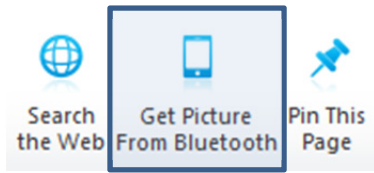


Figure A.17 Get picture from Bluetooth button (Photo by author, 2013)

2. Launch the JBluetoothScanner app on your Android device

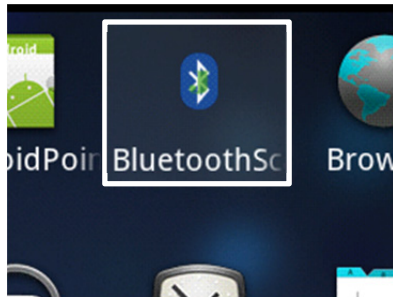


Figure A.18 Android application launcher (Photo by author, 2013)

3. Select the correct device name on the Windows machine

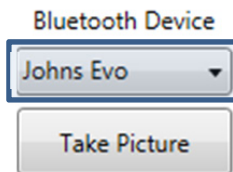


Figure A.19 Bluetooth Scanner device select (Photo by author, 2013)

4. Point the Android at the desired location
5. Click “Take Picture” on the Windows machine

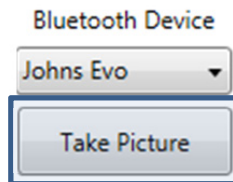


Figure A.20 Bluetooth Scanner take picture (Photo by author, 2013)

6. After a few seconds (about 5), the photo should appear on your Tablet/PC.
7. You can then crop and rotate the picture as necessary. After selecting the Crop checkbox, click and drag from corner to opposite corner on the image to choose the crop area. You can click and drag again to select a new box.



Figure A.21 Bluetooth Scanner image editing tools (Photo by author, 2013)

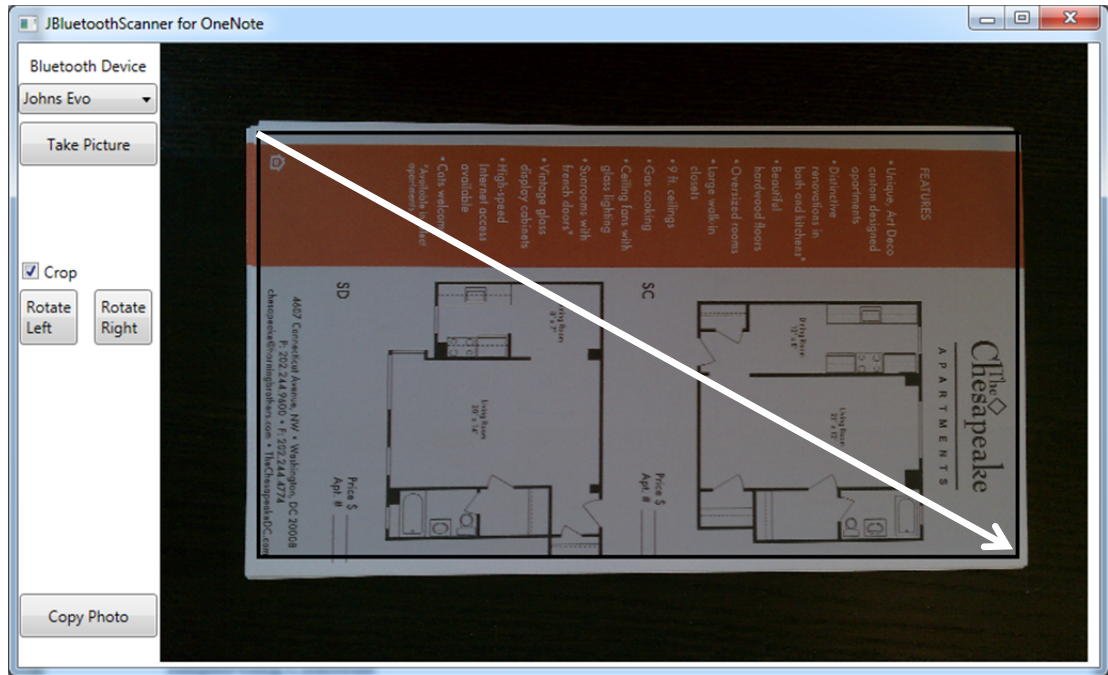
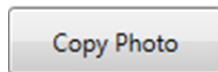


Figure A.22 Bluetooth Scanner interface with cropped photo (Photo by author, 2013)

8. Click Copy Photo to place the photo on your clipboard. You can now paste the photo in OneNote or in any other program as needed.



A.12 Importing Content

There are a number of ways to import content. Note that if using the import wizard all new pages will be imported into the currently open section.

A.12.a PDF to OneNote (Preserving text as text)

1. Double Click on the PDFtoONX.jar file (Requires Java)
2. Select the PDF to convert to the .onx format

3. After a few seconds, a .onx file with the same name as the original document will be created in the same location as the original .pdf file
4. In VText, launch the import wizard from the ribbon.

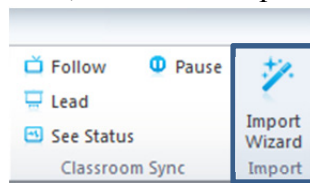


Figure A.23 Method for opening the Import Wizard (Photo by author, 2013)

5. Select the “Import a OneNote XML format document” then click OK

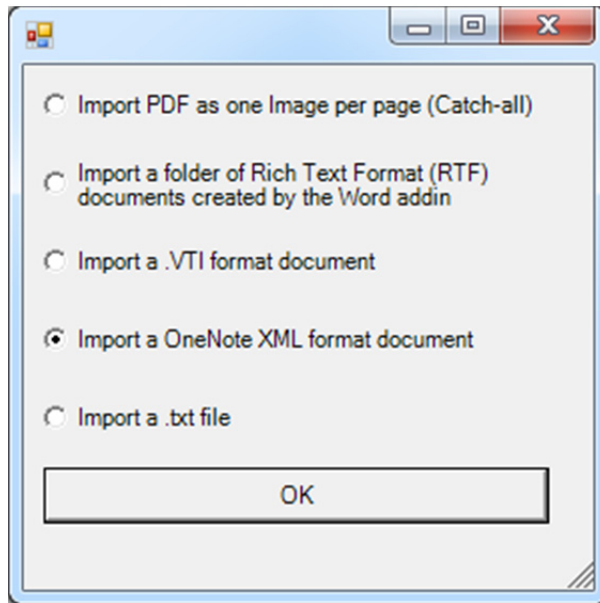


Figure A.24 Import Wizard (Photo by author, 2013)

6. Select the .onx file you created in step 3
7. Choose if you would like VText to add page numbers at the top of each page. Unfortunately these pages numbers are not guaranteed to match the PDF depending on how the PDF is numbered.

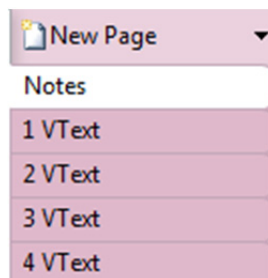


Figure A.25 Results of selecting add page numbers (Used with permission from Microsoft.)

8. Wait for the process to complete.

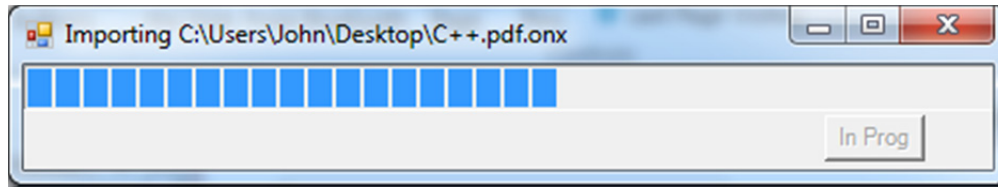


Figure A.26 Example of the Import Wizard Progress Bar (Photo by author, 2013)

A.12.b Import PDF (One image per page)

1. Launch the VText Import Wizard
2. Select “Import PDF as one Image per page (Catch-all)”

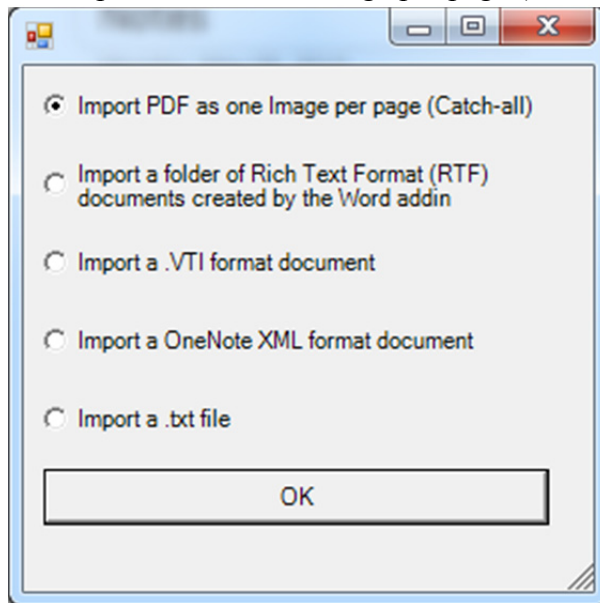


Figure A.27 Import Wizard (Photo by author, 2013)

3. Select the PDF you would like to import
4. Select if you want VText to add page numbers
5. Wait for the process to complete

A.12.c PDF with Watermark Importer

Create the VTI Files

1. Open the PDF to VTI Tool

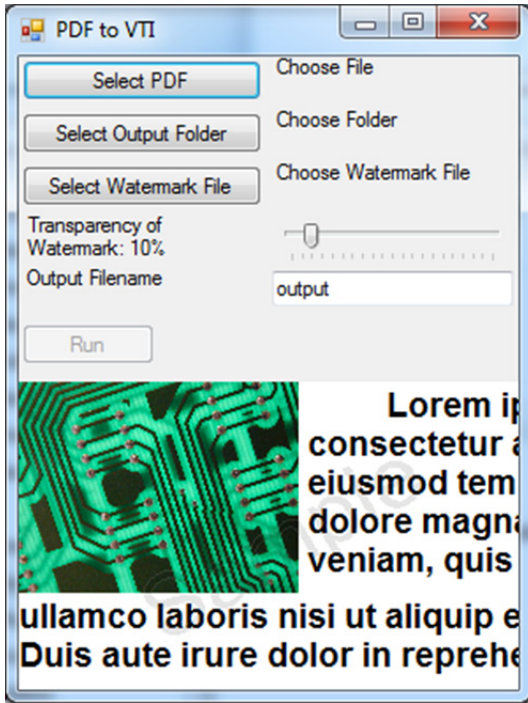


Figure A.28 PDF to VTI Tool (Photo by author, 2013)

2. Click “Select PDF” and choose the PDF you wish all students to use
3. Click “Select Output Folder” and choose the folder where the .vti files should be placed
4. Click “Select Watermark File” and choose the plain text file which contains the various watermarks that should be created. This file should include each watermark desired on a separate line. Using “Notepad” to create this file is recommended.

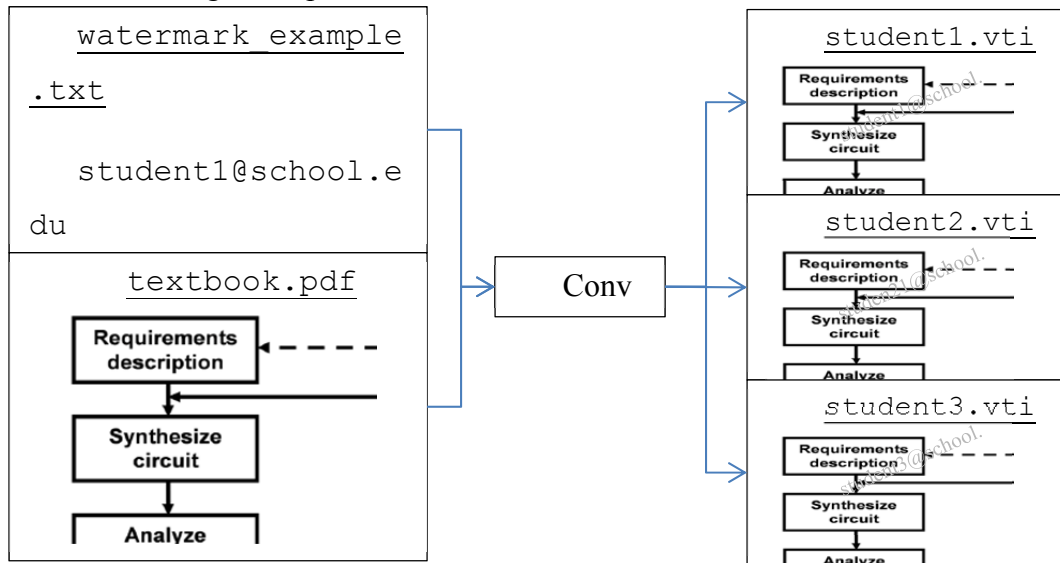


Figure A.29 The PDF to Watermarked VTI

5. Choose the transparency setting as desired. The level of the effect can be previewed in the image at the bottom of the tool's window.
6. Provide a name for the output files. The files will have the following format: <output Filename>_<watermark>.vti
7. Click "Run"
8. Provide the appropriate VTI file to each of your readers.

Importing VTI Files

1. Launch the VText Import Wizard
2. Select "Import a .VTI format document"
3. Click "OK"
4. Wait for the process to complete

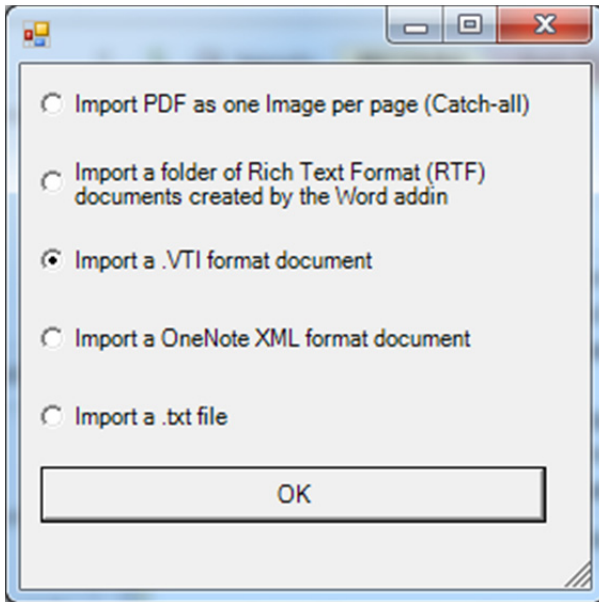


Figure A.30 Import Wizard (Photo by author, 2013)

A.12.d RTF to OneNote

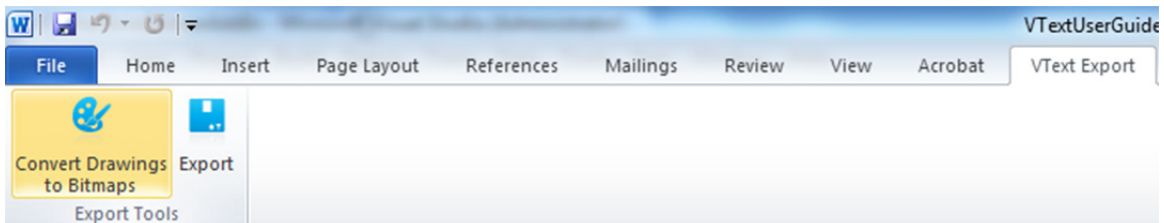


Figure A.31 VText Word Ribbon (Photo by author, 2013)

1. Click on “Convert Drawings to Bitmaps” to convert all drawing canvases to a rasterized image.
2. Click on “Export”
3. Choose the *empty* or *new* folder to store the converted page files
4. In OneNote, open the VText Import Wizard
5. Select “Import a folder of Rich Text Format (RTF) ...”

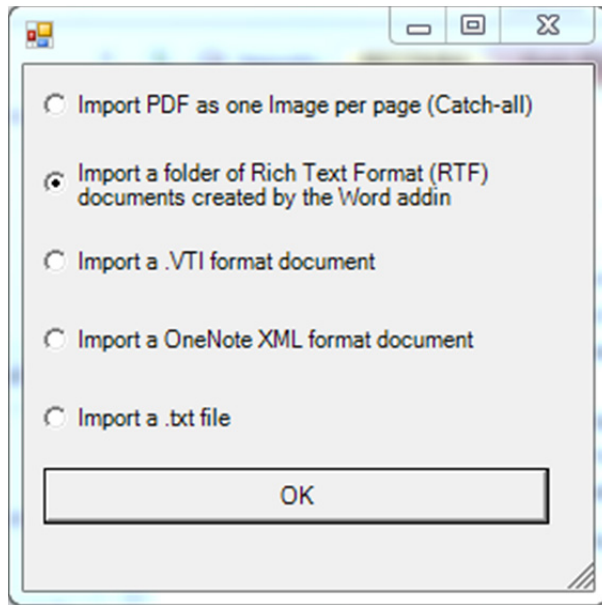


Figure A.32 VText Import Wizard (Photo by author, 2013)

6. Choose the folder you exported into in step 3
7. Let the process finish

A.12.e TXT Importer

1. Open the VText Import Wizard

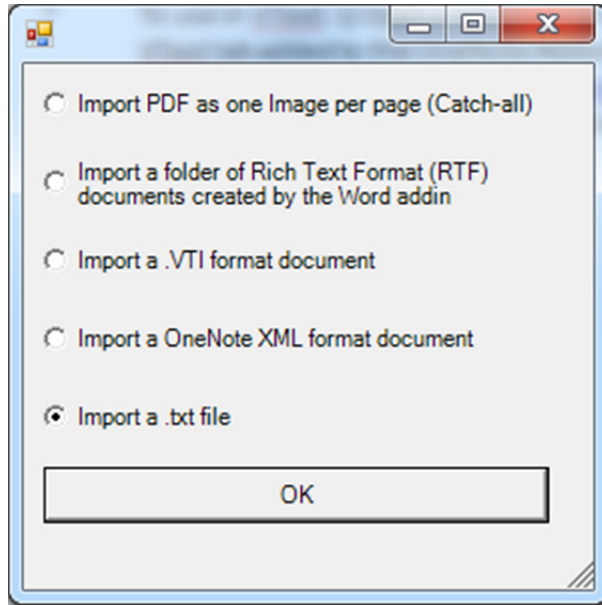


Figure A.33 VText Import Wizard (Photo by author, 2013)

2. Select the text file to import
3. Choose if you'd like page numbers added
4. Let the process finish

A.13 Troubleshooting

Please contact jcristy (ATT) vt.edu. This section will be updated as common issues become known. It is also possible that this document has been updated. Please visit

<http://filebox.vt.edu/users/jcristy/VText> or <http://filebox.ece.vt.edu/~jgtront/etextbook/>

Event	Action
Add-in Crashes <ul style="list-style-type: none"> • The add-in has stopped working • No icons on the VText tab Buttons do nothing 	Unfortunately software under development is prone to crashes. To bring VText back up: <ol style="list-style-type: none"> 1. Close OneNote 2. Open the Windows Task Manager (CTRL+SHIFT+ESC) 3. Open the processes tab <p style="margin-left: 20px;">Note: For steps 4 and 5, if the process is not listed skip to the next step</p> 4. Locate and select ONENOTE.EXE (ONENOTEM.EXE is ok). Click End Process. 5. Locate and select dllhost.exe. Click End Process. 6. Re-open OneNote 7. <i>If the add-in still does not load</i>, Open the Menu (File in Office 2010, the Circle at the top left in Office 2007) 8. Click Options

	<p>9. Select Add-ins from the list on the left of the screen</p> <p>10. Find “Manage: COM Add-ins” click Go.</p> <p>11. If the checkbox next to VText is not selected, select it, and click OK. The add in should now be loaded</p> <p>12. If the checkbox next to VText is selected, unselect it, click OK, then go back to step 7.</p>
Gestures stop working	Unfortunately the touch gestures for VText had to be designed to interact with Windows at a low level. It is hoped that Microsoft will simplify this in the future; however, in the meantime, please close and reopen OneNote.
After page change gesture, I lose my spot on the page	<p>VText should return you to your original spot on the page before changing the page; however, this can stop working from time to time. If so follow these directions:</p> <p>Close and reopen OneNote</p> <p>OR</p> <p>Switch from two page to one page mode and back again, or vice versa.</p>

A.14 Other Resources

- Many out of copyright books are available as .txt (and will work with the .txt importer) at the Project Gutenberg website at <http://gutenberg.org>.
- Some free books are available at <http://filebox.vt.edu/users/jcristy/VText>.

A.15 Release Notes

Version	Notes
0.9.2 alpha	<ul style="list-style-type: none"> • Importers now ready for evaluation use • Added Setting to turn off automatically switching the active window when hovering with the pen • Fixed bug where pen and touch could not use the right-click menu to access “Search the web” and “Take Note” • Added graphic to indicate the user can drag up on the navigation buttons
0.9.1 alpha	<ul style="list-style-type: none"> • Gestures should be more robust • Navigation buttons should no longer open improperly • Pinned Page names load properly when a page title includes a comma • Dragging up from navigation buttons now offers options to pin the current page and view pins • Added ability to rotate images to Bluetooth Scanner • Can now add page numbers when importing .onx • Changed .onx format, old .onx files will no longer work
0.9 alpha	This is the first public release of VText

Note: The version of the addin can be found by clicking on the Version button in the Options group.

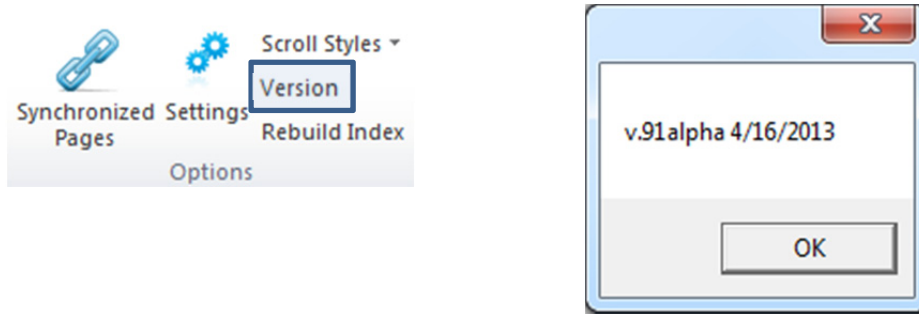


Figure A.34 VText version information (Photo by author, 2013)

Appendix B Pre-Survey Results

Questions 1-4 involved identification information and consent.

5. Question	Not at all	A little	Some	Quite a bit	A great deal	Total Responses	Mean
I used the textbook to collaborate with my classmates.	15	16	18	11	5	65	2.62
I communicated with the instructor by email.	4	18	21	14	8	65	3.06
I completed assignments on time.	0	0	0	8	57	65	4.88
I prepared for an exam.	0	0	3	31	31	65	4.43
I preferred to study alone.	2	5	19	26	13	65	3.66
I preferred to study in a group.	3	17	31	11	3	65	2.91
I procrastinated with reading and/or other assignments.	4	17	22	16	7	66	3.08
I talked to the instructor in person outside of class about ideas and concepts from the readings or classes.	24	27	11	3	0	65	1.89
I took notes in class.	3	4	11	18	29	65	4.02
I used class notes from others.	22	28	12	3	0	65	1.94
I shared my class notes with others.	15	26	17	6	1	65	2.26
I brought a printed textbook(s) to class.	28	11	6	11	9	65	2.42
I reflected on assigned readings before class.	13	23	21	7	1	65	2.38
I brought a laptop or similar device (e.g., netbook) to class.	6	0	2	5	52	65	4.49
I brought a tablet computer to class (iPad or similar).	12	0	2	8	43	65	4.08

6. What percentage of assigned readings do you typically complete for a class in your major (first year students should consider their most recent year in high school)?

Answer	Bar	Response	%
0%		0	0%
1-25%		0.047619	3
26-50%		0.222222	14
51-75%		0.396825	25
76-100%		0.333333	21

7. What percentage of assigned readings do you typically complete for a class outside of your major (first year students should consider their most recent year in high school)?

Answer	Bar	Response	%
0%		0.015873	1
1-25%		0.15873	10
26-50%		0.253968	16
51-75%		0.285714	18
76-100%		0.285714	18

8. About how many hours in a typical 7-day week during the academic year do you read for classes?

Answer	Bar	Response	%
0		0.015873	1
1-5		0.52381	33
6-10		0.222222	14
11-15		0.095238	6
16-20		0.047619	3
21-25		0.031746	2
26-30		0.015873	1

More than 30		0.047619	3	5%
--------------	--	----------	---	----

9. About how many hours in a typical 7-day week during the academic year do you read for pleasure?

Answer	Bar	Response	%	
0		0.370968	23	37%
1-5		0.403226	25	40%
6-10		0.096774	6	10%
11-15		0.096774	6	10%
16-20		0	0	0%
21-25		0.016129	1	2%
26-30		0.016129	1	2%
More than 30		0	0	0%

10. Rate yourself in terms of reading ability compared to your peers at Virginia Tech.

Answer	Bar	Response	%	
Lowest 10%		0	0	0%
Below Average		0.109375	7	11%
Average		0.5	32	50%
Above Average		0.3125	20	31%
Highest 10%		0.078125	5	8%

11. Identify a level of comfort with your use of computer-based technology. Think about how you use and how often you use such things as a personal computer, mobile phone, social networks, Internet, gaming devices, and iPods.

Answer	Bar	Response	%	
--------	-----	----------	---	--

Computer-based technologies frustrate me so I try to avoid using them.	0	0	0%
I am generally uncomfortable with computer-based technology.	0.016393	1	2%
Depending on the technology (hardware or software) I am neither uncomfortable nor comfortable. I fall somewhere in the middle of this scale.	0.04918	3	5%
I am generally comfortable with computer-based technology.	0.508197	31	51%
I am very comfortable and typically seek out the latest hardware and software.	0.42623	26	43%

12. Below are reasons often given for why technology facilitates student learning. For each item, do you agree or disagree?

Question	Disagree	Tend to Disagree	Neither Agree Nor Disagree	Tend to Agree	Agree	Total Responses	Mean
Technology facilitates access to resources.	0	1	0	18	43	62	4.66
Technology lessens administrative tasks for students.	3	4	11	24	20	62	3.87
Technology organizes student work.	0	0	6	24	32	62	4.42
Technology makes students feel more connected to other students.	0	5	16	19	23	63	3.95
Technology enhances communication from student to professor.	0	3	9	18	32	62	4.27
Technology enhances communication from professor to student.	0	1	5	19	37	62	4.48
Technology leads to increased student engagement in learning.	1	2	17	19	23	62	3.98
Technology improves student grades.	4	2	17	21	18	62	3.76

13. I have used e-textbooks in a previous class in high school.

Answer	Bar	Response	%
Yes	0.290323	18	29%

No	0.709677	44	71%
----	----------	----	-----

14. I have used e-textbooks in a previous class in college.

Answer	Bar	Response	%
Yes	0.693548	43	69%
No	0.306452	19	31%

15. I have used or purchased an e-textbook for a class even if the professor did not present it as an option.

Answer	Bar	Response	%
Yes	0.285714	18	29%
No	0.714286	45	71%

16. On a scale of 1-10, please respond the following

Answer	Bar	Response	%	
I am apprehensive to use e-textbooks.	1	0.032258	2	3%
	2	0.032258	2	3%
	3	0.016129	1	2%
	4	0.032258	2	3%
	5	0.129032	8	13%
	6	0.096774	6	10%
	7	0.241935	15	24%
	8	0.16129	10	16%
	9	0.129032	8	13%

I am eager to use e-textbooks	10	0.129032	8	13%
-------------------------------	----	----------	---	-----

17. On a scale of 1-10, please respond the following

Answer	Bar	Response	%	
I am uncomfortable with the idea of using e-textbooks.	1	0.063492	4	6%
	2	0.063492	4	6%
	3	0.047619	3	5%
	4	0.063492	4	6%
	5	0.079365	5	8%
	6	0.063492	4	6%
	7	0.126984	8	13%
	8	0.15873	10	16%
	9	0.142857	9	14%
I am comfortable with the idea of using e-textbooks	10	0.190476	12	19%

18. In addition to the free e-textbook for this course, do you intend to purchase a hard copy?

Answer	Bar	Response	%
Yes	0.209677	13	21%
No	0.354839	22	35%
I already own the hard copy for a previous course	0.354839	22	35%
Not sure at this time	0.080645	5	8%

19. Below are reasons often given for using an e-textbook. Please indicate how important each is to you.

Question	Not Important	Somewhat Unimportant	Neither unimportant or important	Somewhat important	Important	Total Responses	Mean
I can save money.	0	0	3	7	52	62	4.79
It is more convenient to download books.	0	4	5	21	32	62	4.31
I don't have to carry heavy books.	3	2	10	14	33	62	4.16
It will save time studying.	2	2	13	26	20	63	3.95
It will increase my academic performance (i.e., increase grades).	5	3	12	22	20	62	3.79
It will develop skills required for a good job.	4	4	17	19	18	62	3.69
I can share my notes.	2	7	17	19	17	62	3.68
I like to read on a computer.	9	4	12	26	11	62	3.42
It will help me graduate on time.	7	5	21	13	16	62	3.42
I will read more often.	5	10	20	16	11	62	3.29

20. Please list other reasons for using an e-textbook.

Cheaper than hard copy

If an e-textbook is not cheaper than the hard copy, than I prefer to have the hard copy.

Can easily search the book by typing a concept/definition.

it can be highlighted and edited :D

It is easier to find specific topics with ctrl+f, because if we are stuck on a problem, and we do not know how to do one portion of it, we could look it up in the book, but that takes a bit of time because we have to look page to page.

If it can be downloaded I can put it on my kindle.

Ease of use and control+F.

E-textbooks are very convenient

You can search on it very easily.

It is searchable

21. Below are reasons often given for NOT using an e-textbook. Please indicate how important each is to you.

Question	Not at all	A little	Some	Quite a bit	A great deal	Total Responses	Mean
It is more convenient to download books.	5	10	24	9	12	60	3.22
I am easily distracted by other functions on my computer/e-reader (e-mail, Facebook, apps, etc.).	11	9	14	16	12	62	3.15
I prefer to read hardcopy books when I study.	11	10	14	14	13	62	3.13
I want to have access to the book after the course concludes.	11	9	14	15	12	61	3.13
I am afraid of losing material due to technology failure.	17	14	12	14	5	62	2.61
I will read less often.	27	15	12	2	6	62	2.11
The course instructor does not use it.	30	14	9	6	3	62	2
I will get lower grades.	32	15	9	2	4	62	1.89
I don't own a computer or tablet.	56	2	1	3	0	62	1.21

22. Please list other reasons for NOT using an e-textbook.

I dont like reading on the computer. I like to have the book in my hand.

Sometimes I do not have internet, will I still be able to use the book?

Time spent using an online textbook is more time spent in front of a computer screen. I am a computer engineering student, so I try to minimize computer time where I can.

bad for your eyes to read from a screen

too use to hardcopy when studying

It could provide easy access to it during tests with smartphones.

Finding certain sections and passages is harder than flipping through a physical book. I hope it would have a working index

I cannot stand looking at the screen for a long time at all

It is difficult to flip quickly between pages especially in math texts when one would need to view answers and questions at the same time. I feel like i can get lost in an e-book and have no way of reasoning where I am in the text spatially.

It's going to affect my sight. Using laptops a lot in not a healthy thing.

Eye fatigue while looking at computer

23. Thinking about your use of an e-textbook(s) for this course, how do you anticipate this format might meet each of the following learning needs?

Question	Not at all	A little	Some	Quite a bit	A great deal	Total Responses	Mean
E-textbooks will distract me from the ideas and concepts taught in this course.	29	19	8	3	2	61	1.85
E-textbooks will allow me to better organize and structure my learning.	5	15	18	14	9	61	3.11
E-textbooks will decrease my engagement with course content.	30	24	5	1	1	61	1.67
E-textbooks will offer greater flexibility to learn the way I want.	4	16	18	16	7	61	3.1
E-textbooks will help me interact and collaborate more with classmates.	13	23	13	8	4	61	2.46
E-textbooks will make my study time less efficient.	29	23	7	0	2	61	1.74
E-textbooks will allow me to interact more with my professor.	17	22	14	7	1	61	2.23
E-textbooks will enhance my learning.	7	21	20	7	7	62	2.77

24. What interests you about using e-textbooks?

Cheaper

The lower cost and not having to carry a heavy textbook around

inbuilt tutoring

Their convenience and lower costs.

Its nice to save some money on textbooks

Its nice not to carry around a heavy textbook

no heavy books

Nothing

Price

Cheaper Cost

You will always have it in your computer

Always having a copy of the book on hand.

Convenience, where if I leave town and forget my book, I could always re-download the book and read what I missed.

I can access it whenever I'm on my computer without needing to carry around a heavy textbook.

free

I take notes on my computer why not be able to read on my computer also.

They are convenient, and carry less weight in the back pack.

It might make things easier to search for in the book

The price.

Flexibility

Saving money.

I only have to carry my tablet to have all my textbooks, and this is cheaper than hard-copies.

I can easily search e-textbooks for key terms and concepts.

Being able to search text and concepts

That they can be carried easily since I always carry my computer with me.

25. What concerns you about using e-textbooks?

It's hard to read and reference back to material in.

Not being able to sell the book back to get my money back

The ease of use and accessibility of the e-textbook.

I don't think I will focus in class

I don't like reading 20 pages on the computer, it hurts my eyes

I get easily distracted on the computer.

It's on a computer, I like hard copies

Nothing

I do not do well reading from a computer

If your computer dies, you can't access your book

Some times the book is only viewable on a website with some sort of protection of the text. It is much easier if I can get a pdf/mobi/epub of the book rather than having to rely on some website to host my book until the end of time. Many publishers go through amazon kindle so the books are easily viewable offline using the various platforms the kindle app is made for.

It might distract me at times when I see a notification on my laptop.

The transition to a paperless world

potential licensing issues/ access when the internet is down

hard to highlight and i find it easier to remember when you can touch the book compared to a screen

Not having the textbook after the class.

I'm afraid of technology errors and that I won't be able to utilize the book properly.

Not having access after the course concludes.

My eyesight

Reading at a computer for long periods

I don't like that they have subscription times that end after you finish the class.

Will they expire after a while? I would like to reference previous textbooks in future classes.

I often take notes on my computer so reading an e-textbook would be hard in that case since a laptop screen is small. If there was a note-taking system on the e-reader, that would mitigate this problem as long as that feature can be accessed independently of viewing a certain page in the book.

reliability

Not easy to read on a tablet computer (fujitsu)

That I will not be able to make notes near the text in the book, such as underline.

Appendix C Post-Survey Results

Questions 1-4 involved identification information and consent.

5. What percentage of assigned readings did you complete for this class?

Answer	Bar	Response	%
0%	0.037037	1	4%
1-25%	0.259259	7	26%
26-50%	0.407407	11	41%
51-75%	0.111111	3	11%
76-100%	0.185185	5	19%

6. Compared to before the trial period...

Answer	Bar	Response	%
I read more	0.222222	6	22%
I read about the same	0.703704	19	70%
I read less	0.074074	2	7%

7. Thinking about the amount of reading you completed during the trial period, was this because you were using or could use VText?

Answer	Bar	Response	%
--------	-----	----------	---

Yes	0.37037	10	37%
No	0.62963	17	63%

8. Engagement means student time and energy devoted to educationally purposeful activities and is indicated by such things as student-faculty contact, cooperation among students, active learning, prompt feedback, and time on task. Compared to other courses and before the VText trial,

Answer	Bar	Response	%
I was more engaged in the course	0.153846	4	15%
I was engaged about the same in the course	0.846154	22	85%
I was less engaged in the course	0	0	0%

9. Did you have technical issues with VText?

Answer	Bar	Response	%
Yes	0.111111	3	11%
No	0.888889	24	89%

Note: Question 20 appeared directly after 9 and before 10 when presented to the students.

20. If yes, what issues did you have?

I only got chapter 8

Bronson text seemed to only start late in the book. The earlier chapters didn't seem to be present.

Uninstalling the old to reinstall the new

10. Did you purchase, rent, or borrow a hardcopy of the textbook?

Answer	Bar	Response	%
--------	-----	----------	---

Yes	0.851852	23	85%
No	0.148148	4	15%

11. At what point during the semester did you purchase a hardcopy?

Answer	Bar	Response	%
At the beginning	0.958333	23	96%
After the first month of class	0.041667	1	4%
Somewhere in the middle of the semester	0	0	0%
Near the end of the semester	0	0	0%

12. Thinking about your level of engagement in this course, was this because you were using VText?

Answer	Bar	Response	%
Yes	0.185185	5	19%
No	0.814815	22	81%

13. How have your feelings toward eTextbook use changed since before the trial period? Are you more or less interested in using them in the future?

Answer	Bar	Response	%
Much Less	0	0	0%
Less	0	0	0%
Somewhat Less	0	0	0%
The Same	0.518519	14	52%
Somewhat More	0.222222	6	22%
More	0.222222	6	22%
Much More	0.037037	1	4%

14. Did you use OneNote during this course (For any reason not just with the textbook)?

Answer	Bar	Response	%
Yes, before the VText trial	0.555556	15	56%
Yes, but only during the VText trial	0.259259	7	26%
No	0.185185	5	19%

15. What was the primary way you read the e-Textbook (Bronson or Schildt's C++, A Beginner's Guide)?

Answer	Bar	Response	%
I used VText	0.481481	13	48%
I used just the provided OneNote file (C++, A Beginner's Guide), but did not install VText	0.444444	12	44%
I didn't use VText or OneNote	0.074074	2	7%

16. If you used the C++, A Beginner's Guide e-Textbook in VText or OneNote, how satisfied were you with the presentation?

Answer	Bar	Response	%
Very Dissatisfied	0	0	0%
Dissatisfied	0	0	0%
Somewhat Dissatisfied	0.074074	2	7%
Neutral	0.296296	8	30%
Somewhat Satisfied	0.259259	7	26%
Satisfied	0.296296	8	30%
Very Satisfied	0.074074	2	7%

17. If you used the Bronson e-Textbook in VText, how satisfied were you with the presentation?

Answer	Bar	Response	%
Very Dissatisfied	0	0	0%

Dissatisfied	0.041667	1	4%
Somewhat Dissatisfied	0.083333	2	8%
Neutral	0.458333	11	46%
Somewhat Satisfied	0.166667	4	17%
Satisfied	0.125	3	13%
Very Satisfied	0.125	3	13%

18. What did you find was the top advantage to using an e-textbook (check all that apply)?

Answer	Bar	Response	%
Cost	0.703704	19	70%
Portability	0.777778	21	78%
Sharing notes among classmates	0.111111	3	11%
Saved time studying	0.222222	6	22%
Access to Web	0.333333	9	33%
Ease of acquiring materials	0.444444	12	44%
Other	0.037037	1	4%

19. What did you find was the top disadvantage to using an e-textbook (check all that apply)?

Answer	Bar	Response	%
Prefer to read hardcopy books	0.592593	16	59%
Afraid of losing material due to technical failure	0.333333	9	33%
Don't own a compatible device	0	0	0%
Course instructor does not use it	0.074074	2	7%
Distraction of other functions on the computer (e-mail, Facebook, etc.)	0.666667	18	67%

21. Which features of VText and OneNote did you use? How satisfied were you with each?

Question	Very Dissatisfied	Dissatisfied	Somewhat Dissatisfied	Neutral	Somewhat Satisfied	Satisfied	Very Satisfied	Did not use	Total Responses	Mean
Search the contents of the book	0	0	1	3	2	9	5	7	27	4.7
Annotated/Highlighted text in the book	0	0	0	6	2	6	3	10	27	4.352941
Side by side pages	0	0	0	5	3	7	3	9	27	4.444444
Touch Gestures	1	0	1	4	4	2	6	9	27	4.222222
Navigational Buttons	1	0	0	5	5	4	4	8	27	4.157895
Bluetooth Scanner	0	0	0	8	0	0	1	18	27	3.333333
Search the Web	0	0	0	5	1	5	3	13	27	4.428571
Pinned Pages	0	0	0	8	0	1	3	15	27	3.916667
Notes Page	0	0	0	6	3	1	4	13	27	4.214286
Import Wizard	0	0	1	10	1	1	2	12	27	3.533333
User Guide	0	0	1	9	0	1	2	14	27	3.538462

22. How important are these features of VText and OneNote to you, regardless of if you had a chance to use them?

Question	Not at all Important	Very Unimportant	Somewhat Unimportant	Neither Important nor Unimportant	Somewhat Important	Very Important	Extremely Important	Total Responses	Mean
Search the contents of the book	0	0	0	0	2	13	11	26	6.35
Navigational Buttons (ability to flip to the	0	0	1	2	8	11	5	27	5.63

previous and next pages with one small button)									
Pinned Pages (Bookmark pages in the book for later review)	1	0	1	3	8	9	5	27	5.37
Side by side pages (Can see two pages on the screen at one time)	0	0	1	3	13	5	5	27	5.37
Touch Gestures (ability to change pages or pin pages with quick gestures)	1	0	2	2	9	7	6	27	5.33
Annotated/Highlighted text in the book	1	0	3	1	7	10	5	27	5.33
Notes Page (Copy content from the book onto a notes page)	1	1	1	3	12	3	6	27	5.11
Search the Web (Search the web for keywords selected in the text)	1	0	1	6	9	6	3	26	5
Import Wizard (Import PDF's, RTF's and text files into OneNote)	1	1	2	7	4	10	2	27	4.85
Bluetooth Scanner (ability to immediately take a picture with your phone and have it on your Windows Laptop)	1	0	8	7	5	5	1	27	4.26

23. What features did you think were missing?

Could've added titles for each page. OneNote seems like a bad platform for a textbook originally published in print.

It needs to turn off some features when note reading a book. It was frustrating when I would take notes and it would sense gestures and other popups