MFA Review: Project One

Aplia Text

Web-Based Application
Project Developed Outside of SCAD
2008-10

Artie Kuhn
Savannah College of Art and Design
MFA 45-Hour Review

Aplia Text: Table of Contents

Overview.....3

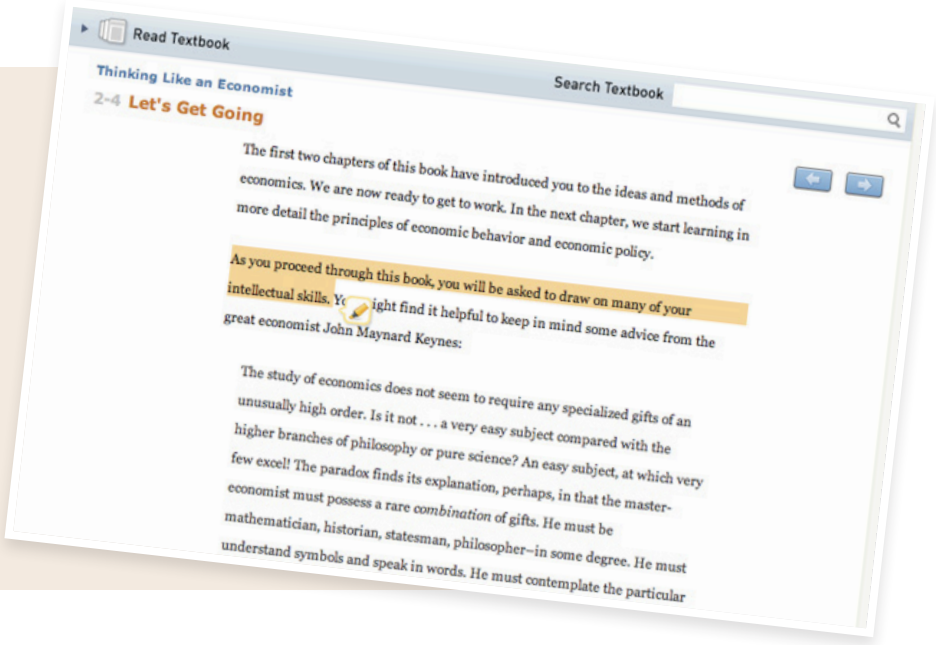
Design Narrative.....4-8

The Problem.....4

The Solution.....5-6

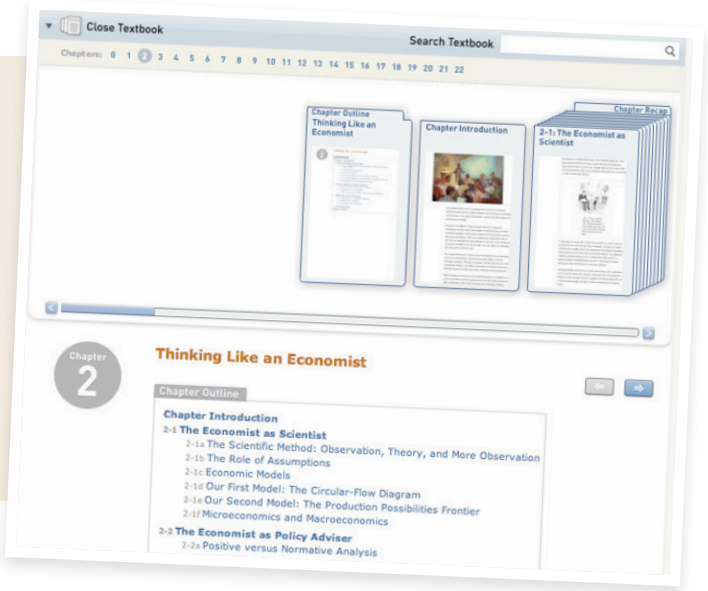
Results.....7-8

Process.....9-16



Aplia Text: Overview

Innovative eBook built on user-centric design principles. Navigation within the web-based application used an innovative “flipbook” that used concepts from data visualization to improve the reading experience. Intuitive interface enabled students to quickly highlight key passages.



Aplia Text: The Problem

Aplia was successful as a start up specializing in rich online homework. Aplia had partnered with textbook publishers to deliver flat textbooks in Flash Paper. They were acquired by the textbook publisher I worked for. Some time after the acquisition, I was asked to join a team to develop a better way to deliver the textbook content within the Aplia platform.

I lead our team in two research initiatives. The first was to engage in ethnographic research to understand how a textbook fit into students' workflow. The second was to analyze the usage data of our existing online textbook platforms and look for usage trends.

After we surveyed and interviewed students from a wide variety of disciplines, we realized that the traditional eBook approach of trying to simulate the physical book was the wrong approach. Instead, we realized that we needed to design an eBook experience that fully leveraged the benefits of an online environment rather than just be a static book on a screen. This meant we would need to rethink the look and feel of each title.

The usage analysis provided one key insight: students do not read online in a linear fashion and then do their associated homework. Instead, most began by answering as many questions as they can of their homework. They then look through their textbook to round out their knowledge. We summarized students as "doers, skimmers and scanners". This was confirmed in our interviews and "shoulder surfing" of students. Because of this, we came to the conclusion that components, like a table of contents, simply wouldn't solve the problems students were having with finding relevant course material, nor was it very engaging.

We also noticed a puzzling contrast in how students used highlights in physical books vs digital books. We noticed time and time again that students that used physical books highlighted their material like crazy. Some even went so far as to chose used books which had been highlighted by their previous owners. However, when we looked at usage patterns in other digital books, we saw an exceptional low number of highlights or notations. After shoulder surfing again, we realized that highlighting in most eBooks is not intuitive or useful and, thus, not common.

After this research phase was complete, our multi-sight team met in a week-long IDEO-style summit where key stake holders gathered to review research, identify the problems to solve and propose solutions. Soon after the summit, I was chosen to lead the group in the creation of Aplia Text. This meant everything from assisting with and testing the UX, art direction of the UI, approving overall development architecture and craft the front-end HTML/CSS and Flash-based components.

Aplia Text: The Solution

I began by thinking, with our team, through the high-level problems that we wanted to solve in the ebook platform: develop a way to merge the aesthetics of the Aplia product with the aesthetics of the individual books, create a UX that helped students be more effective and efficient in their reading of the material and dramatically improve usage of highlighting compared to other digital books.

Visually speaking, we strove for a simple and elegant style. We used Jacob Nielson's research into readability. Therefore, the readline of the copy is quite narrow and the leading is quite large making it easier to read than most online environments. The smaller readline allows for generous margins on both sides of the page which provides plenty of room for larger figures and tables.

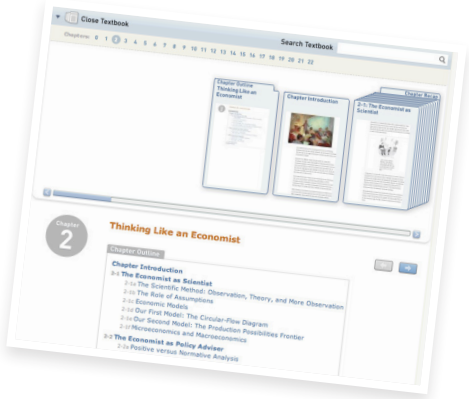
For the UX, we knew we needed to leave many of the navigational elements in physical books behind. We began by completely abaondoning the idea of page numbers (realizing that they were irrelevant in a digital environment) and, instead, broke the book down into chapters, major sections and subsections. This breaking down into logical sections helped students understand how what they were reading fit into the larger scheme of the content. We repurposed an ebook platform owned by our parent company which we felt did an excellent job of breaking textbook into these logical chunks. It was also built in such a way that we could apply our own UI and additional functionality on top of it without too much rework.

After realizing we didn't need to use the physical book as a metaphor, we added a variety of handy features. First, rather than have a glossary, key terms became links within readings that launch small definitions that appear within the page. Those key terms were aggregated to the end of the chapter and presented as key term flash cards and small matching quizzes. Similarly, instead of having answers to selected questions added as an appendix that students would have to jump to, we added a small bar to each question with an answer that toggled it on and off. We scoured our parent publishing company for multimedia and decided that instead of linking out to various multimedia assets (as most competing ebook platforms did), we would embed the assets directly into the page, right at their point of use. In our research, we found that students were highly unlikely to log onto ancillary websites for course materials, but would more frequently use the pedagogically valuable multimedia assets if they were embedded directly in the context.

Time and time again, we had found that students had a hard time navigating through and keeping track of where they were in an eBook. First, we added chapter-level table of contents (which we called "Chapter Outline") to the first page of each

Aplia Text: The Process and Solution (continued)

chapter. This helped students make a mental map of how the different sections of the chapter related to one another.

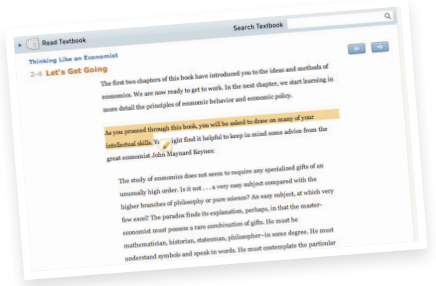


Next, we started thinking about how we could visually display where students were in their readings. After multiple iterations on the problem, I was inspired by Tufte's concept of small multiples and created an innovative flipbook navigation widget. Inspired by the "album view" in iTunes, it allowed students to flip through thumbnails from the book by scrubbing through the pages. "Stacks" of pages on the left and right side gave students a sense of where they were in the chapter. Rolling over individual pages revealed key concepts found on that page. Beyond the textbook component in Aplia, we decided to leverage it throughout Aplia to enable students to jump in and out of their textbook while they were doing their homework. Beyond conceptualizing and designing the widget, I also built the Flash front-end and defined the XML schema that our developers used to power it.

Again, we had seen a huge disparity between use of highlighting in physical book and digital books. When we looked into other eBook platforms we saw that either entire blocks of text are highlighted via a toggle (i/e highlight the entire paragraph or sentence), or required a significant number of clicks and decisions to turn on the highlight tool. Instead, we wanted something that was extremely granular and fast.

I realized that simply selecting text was an implicit statement by the user that they found that text important. So, rather than asking them to highlight that entire paragraph or activate certain highlighting tools, we could simply offer them the option to highlight whenever they selected text. Inspired by radial menus, I designed a simple interface that popups a button a few pixels to the right of the mouse whenever the user highlights text.

Once we had done some initial user testing on my mockups, I created a proof of concept using JavaScript. An initial concern from our QA team and engineers was that we wouldn't be able to build something like this that was cross-browser compatible. My proof of concept laid those fears to rest.



We also realized that we could improve highlights in physical books by aggregating all the student's highlights into one place. We decided to aggregate them all into the end of chapter material. Within the end of chapter page, we simply listed all of the highlights from that chapter along with links back to the page. We also have a link to a page where we list all the paragraphs that contain highlights in order to help students review.

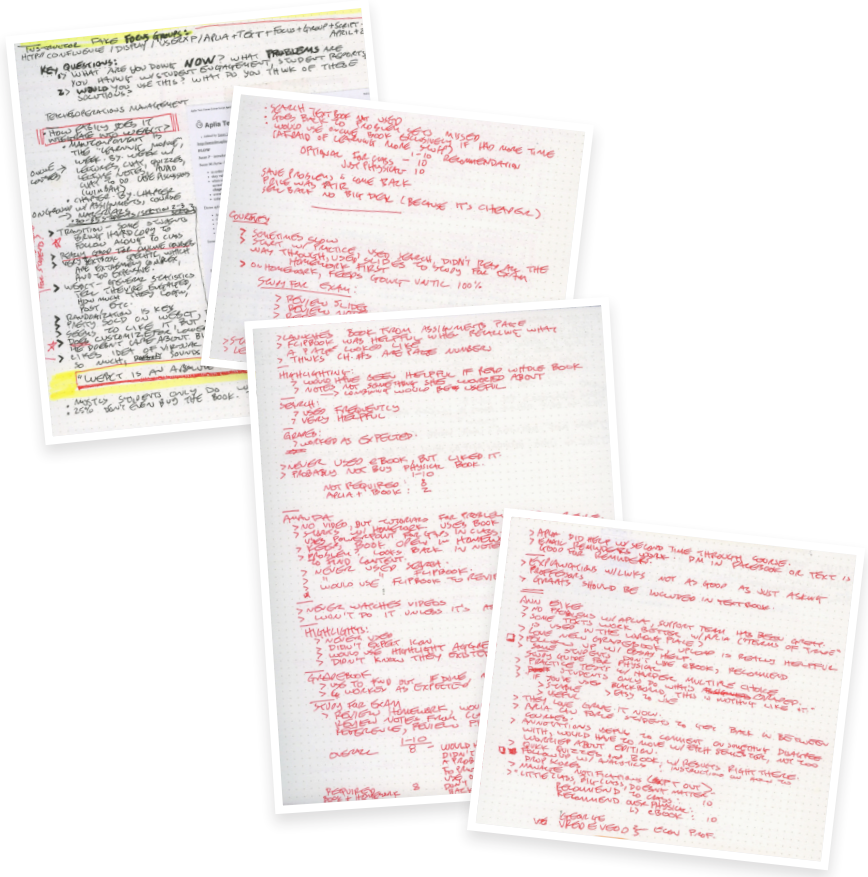
Aplia Text: Results

After a soft launch in summer 2009, Aplia Text was launched as the eBook solution for 24 Cengage textbook titles with over 100,000 students fall 2009. Initial response from both students, instructors and internal stakeholders was been extremely positive and we saw a huge increase in usage the following year. Iterative improvements were made in the design based on feedback we recieved in post-launch interviews.

In research done by other members of our parent company, Aplia Text was consistently ranked as the best eBook offering by students. Students also ranked the flipbook navigation widget as one of their favorite features within all of Aplia.

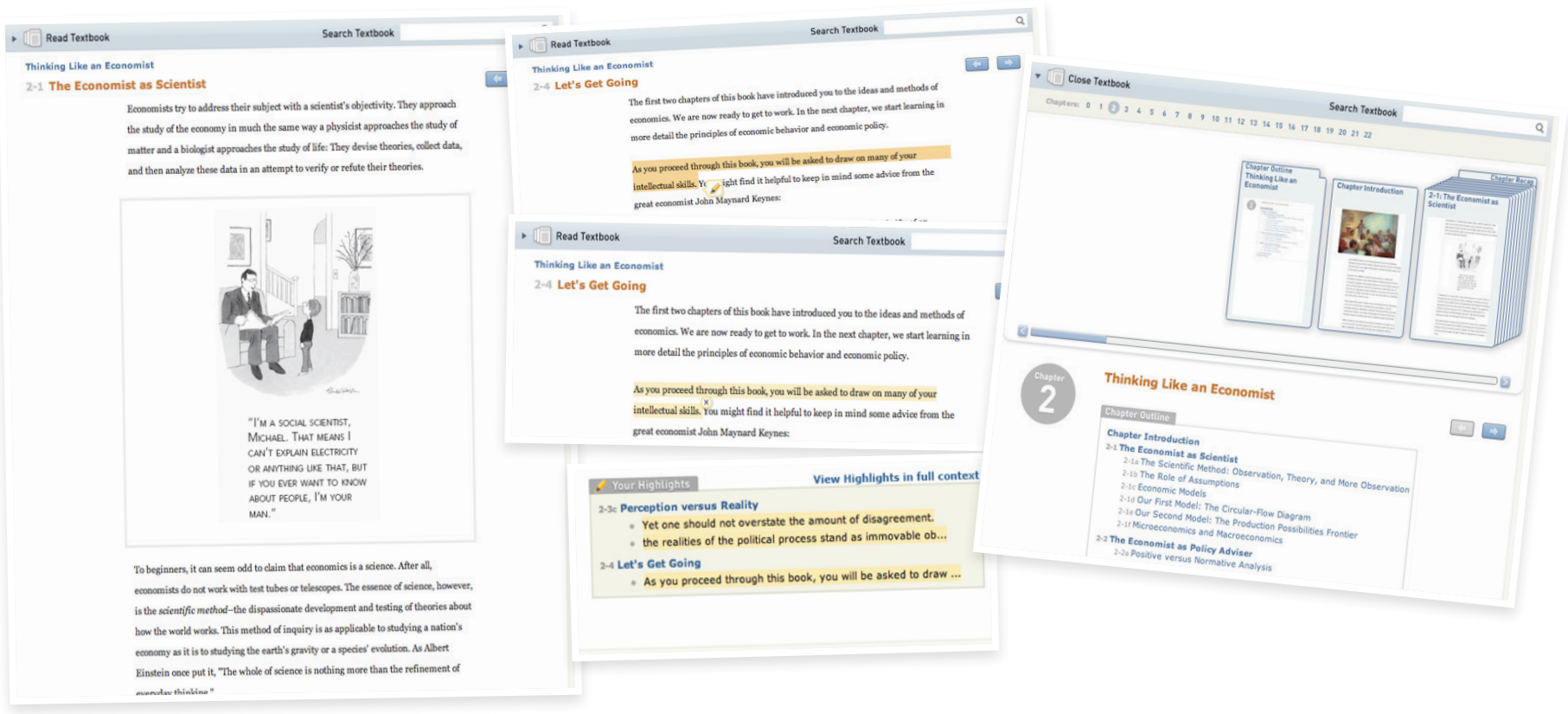
Usage for highlights in Aplia Text was well beyond our expectation. We saw 100x more usage of highlights than any other eBook we have usage data for. In our followup surveys and research with students, highlights ranked near the top of favorite features.

A year later, we scaled our title development process and greatly increased the number of titles offered.



Notes from followup interviews.

Aplia Text: Results

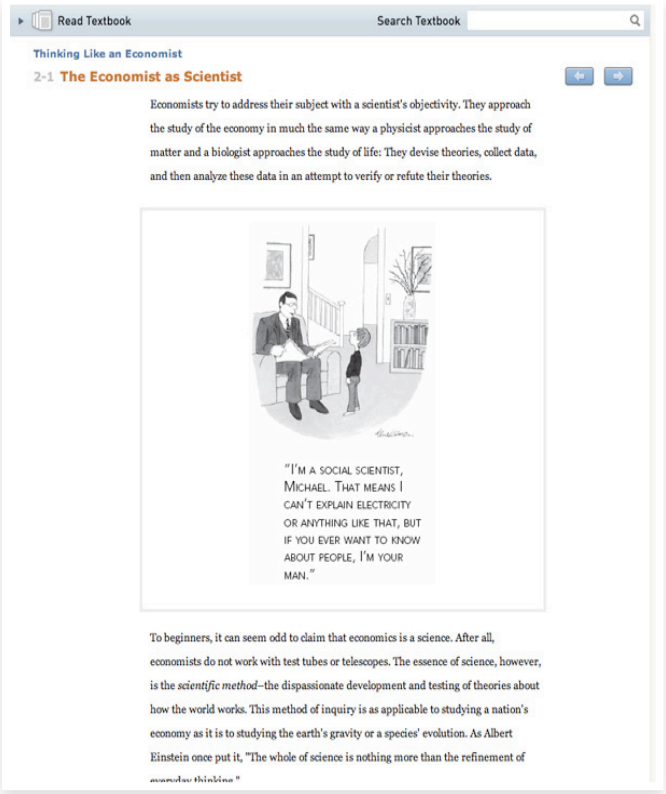


Final screenshots.

Demo video: http://www.youtube.com/watch?v=RYVty91hhIY

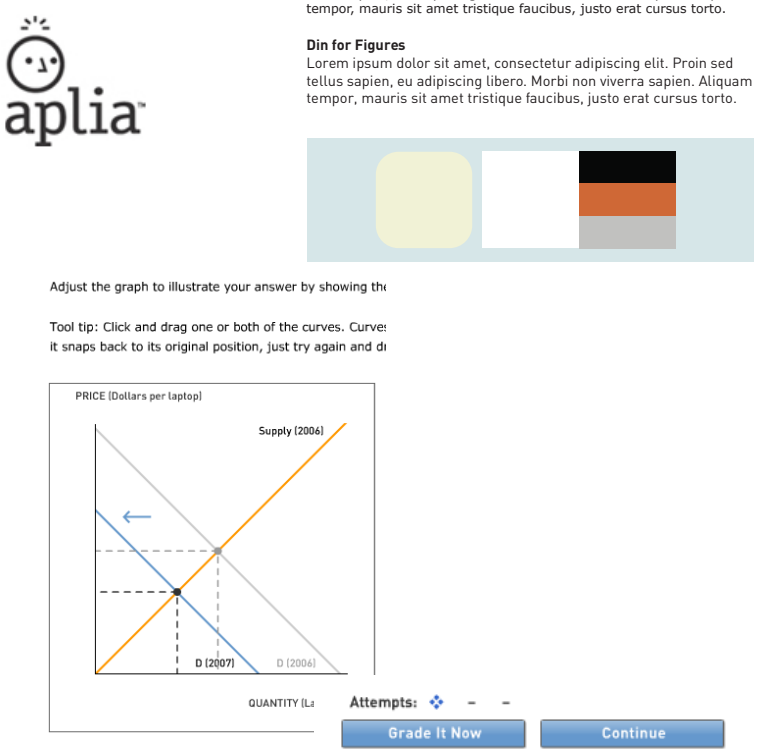
Live demo (follow link and click "View a chapter of the digital textbook"): http://aplia.com/demo?id=124

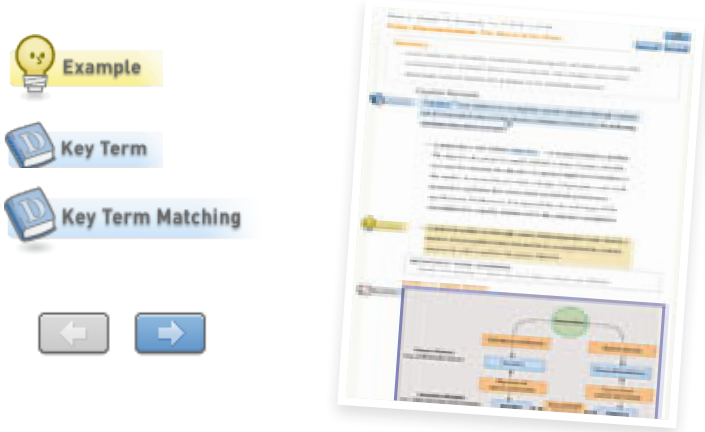
Aplia Text: Process



This is the original platform we chose to build ApliaText on top of. Breaking the content into logical chunks rather than by physical book page number was one of the things we were looking for in a starting point. I also liked the fact that this product already had an established workflow to move content from InDesign files (used for the print book) into XML. While the output HTML was anything but clean or semantic, the underlying XML was extremely semantic and easily mapped the HTML entities and classes we knew we would need (final design on the right).

Aplia Text: Process

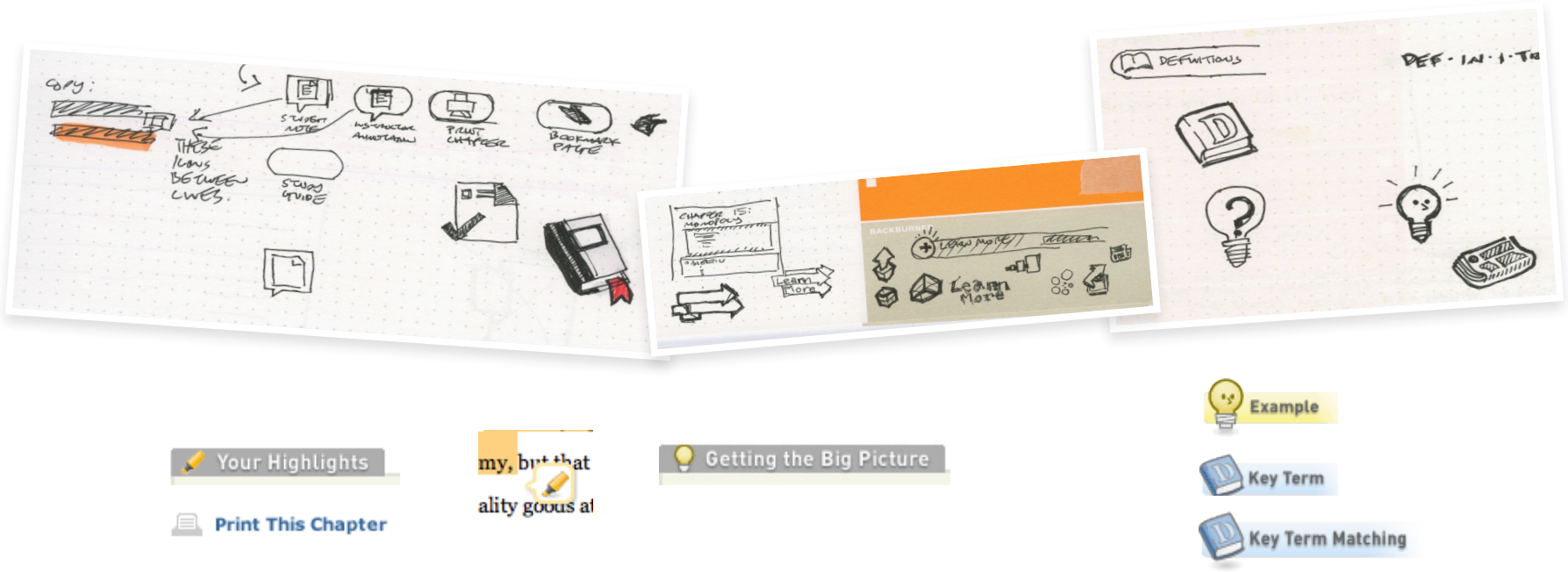




Aesthetically, I wanted to thread a needle. Aplia already had a strong brand, but the font used in the Flash-based homework (Verdana and Din) didn't convey the feel I wanted the book to have (and Din wasn't web-safe). So, I used the branding from Aplia (left) as a launching pad for our own look and feel.

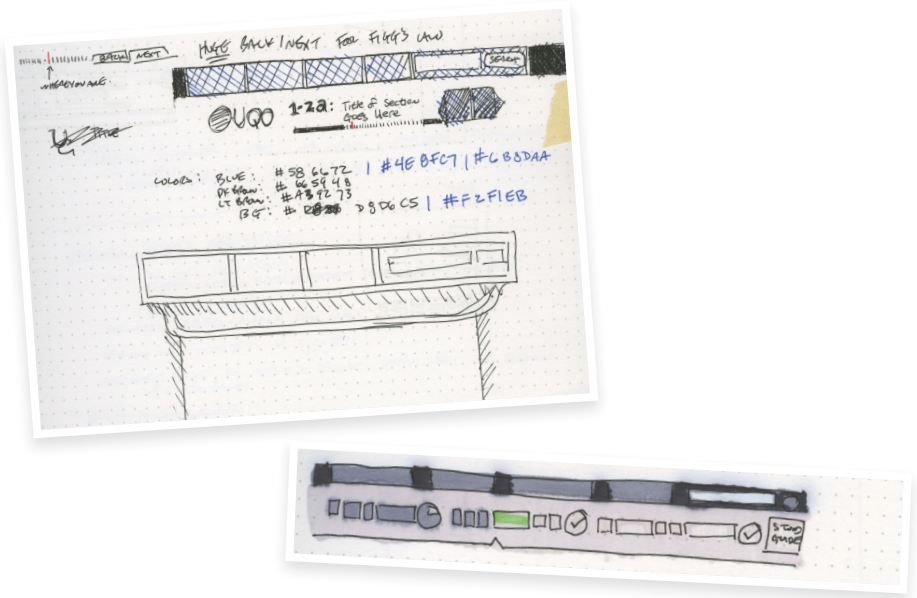
Typeface, size, leading and color were all informed by a series of readability studies Jakob Nielsen had recently released. I even went so far as developing a series of icons to differentiate content. However, because of how they conflicted with book-specific styling elements (far right), these ultimately had to be dropped.

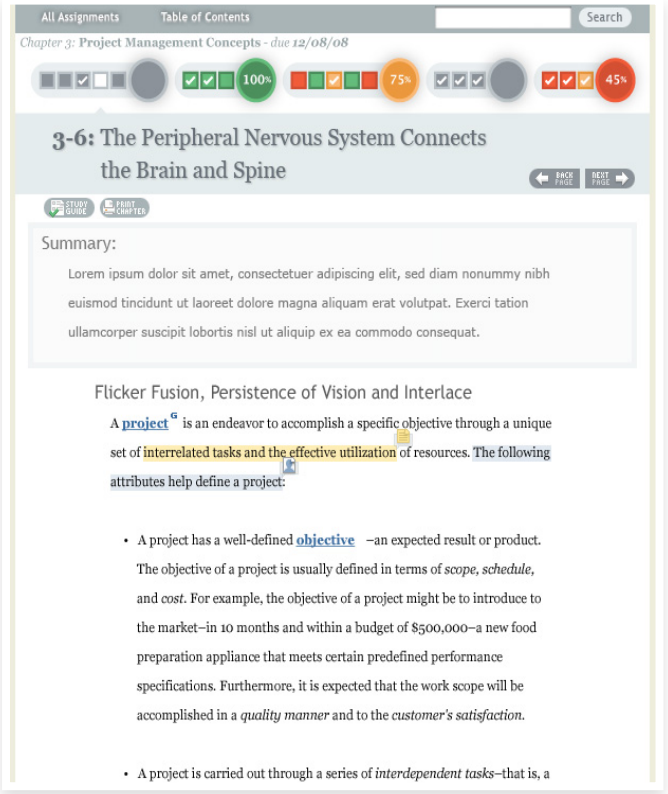
Aplia Text: Process



Icon sketches and final results

Aplia Text: Process

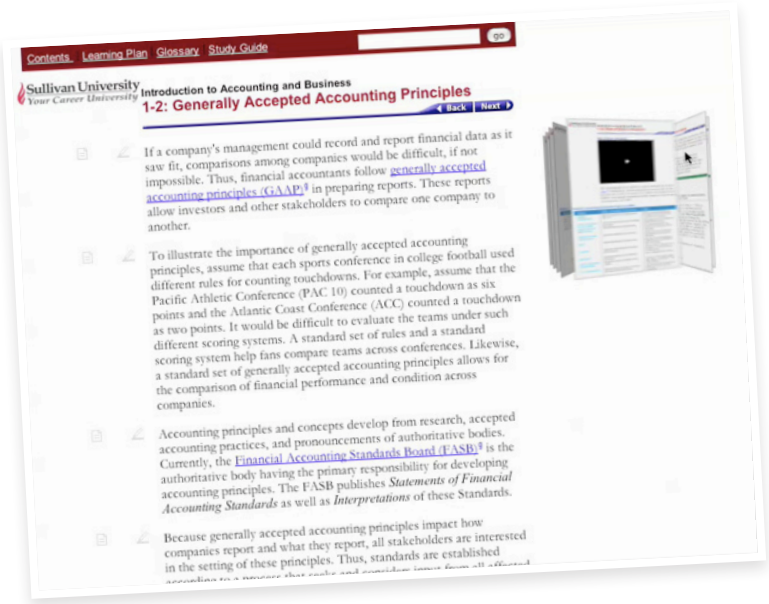


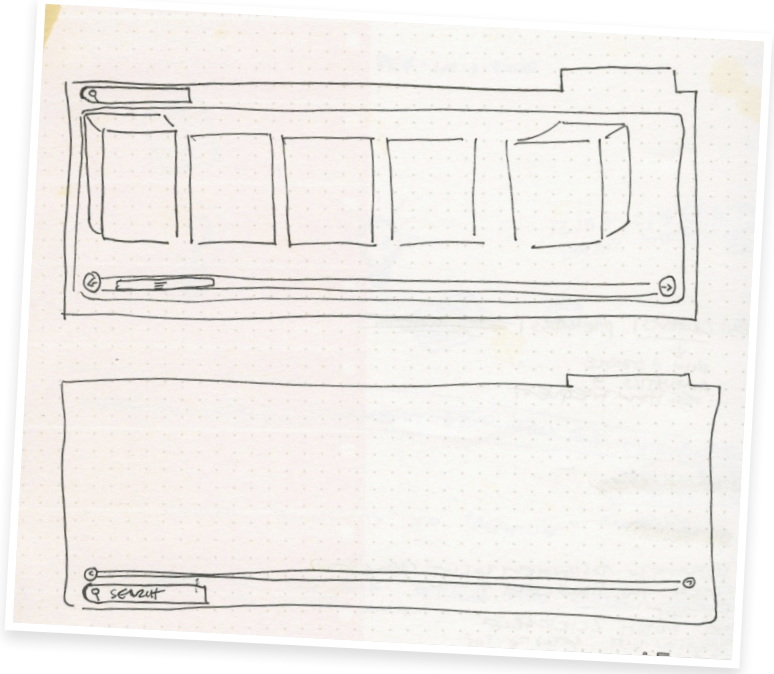


Initially I wanted to integrate the Aplia homework into the book. Students would read some of the content and then complete a short assignment and repeat the cycle through a chapter. The pages would be color coded (an idea I picked up from a Tufte seminar) based on which sections the student needed to review.

However, once we analyzed how instructors assigned homework and readings (they often jumped around in the book) and how students read content (they rarely read linearly), we realized this model would become increasingly cumbersome.

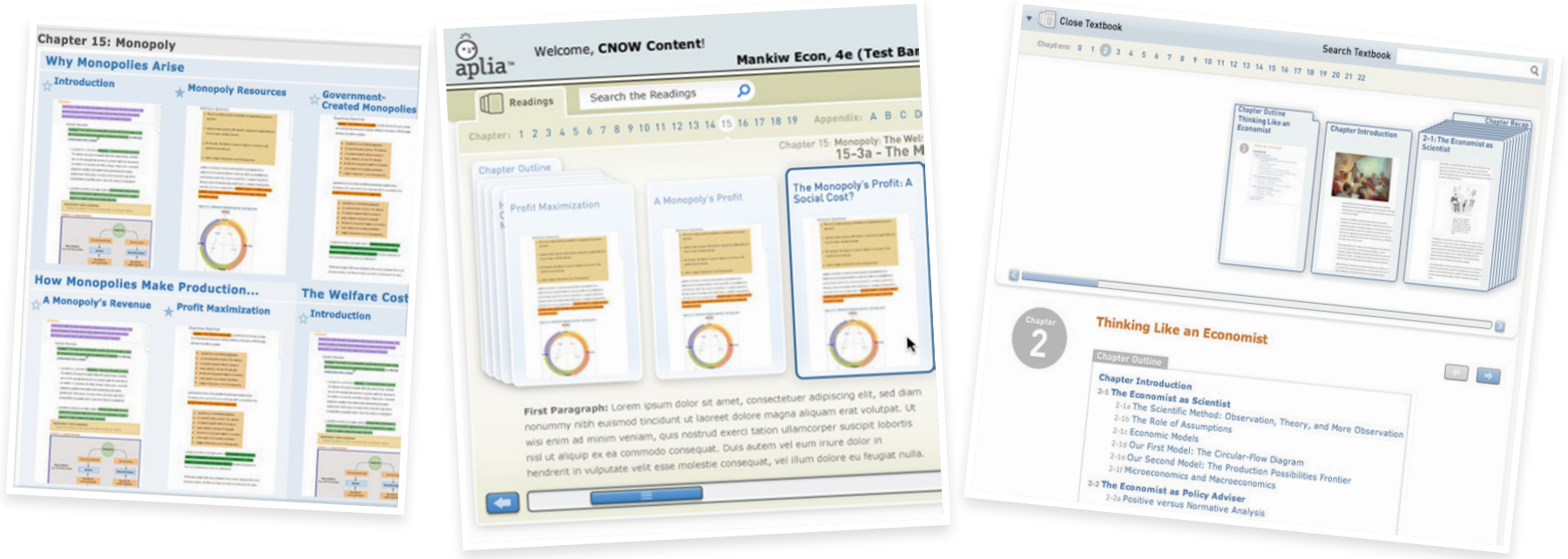
Aplia Text: Process





From the previous iteration, I really liked how one got a sense of where they were in a given chapter. During a team brainstorming session, we recalled how students often flipped through pages. We realized this might make an excellent navigational widget. I quickly mocked up an animated version in After Effects (far left) and showed it to the team. We all agreed it was the direction we wanted to head. I began iterating on this "flipbook" concept. I wanted the small thumbnails to carry meaning, so I played with highlighting passages based on the content type. I liked it conceptually, but it was too much visual noise.

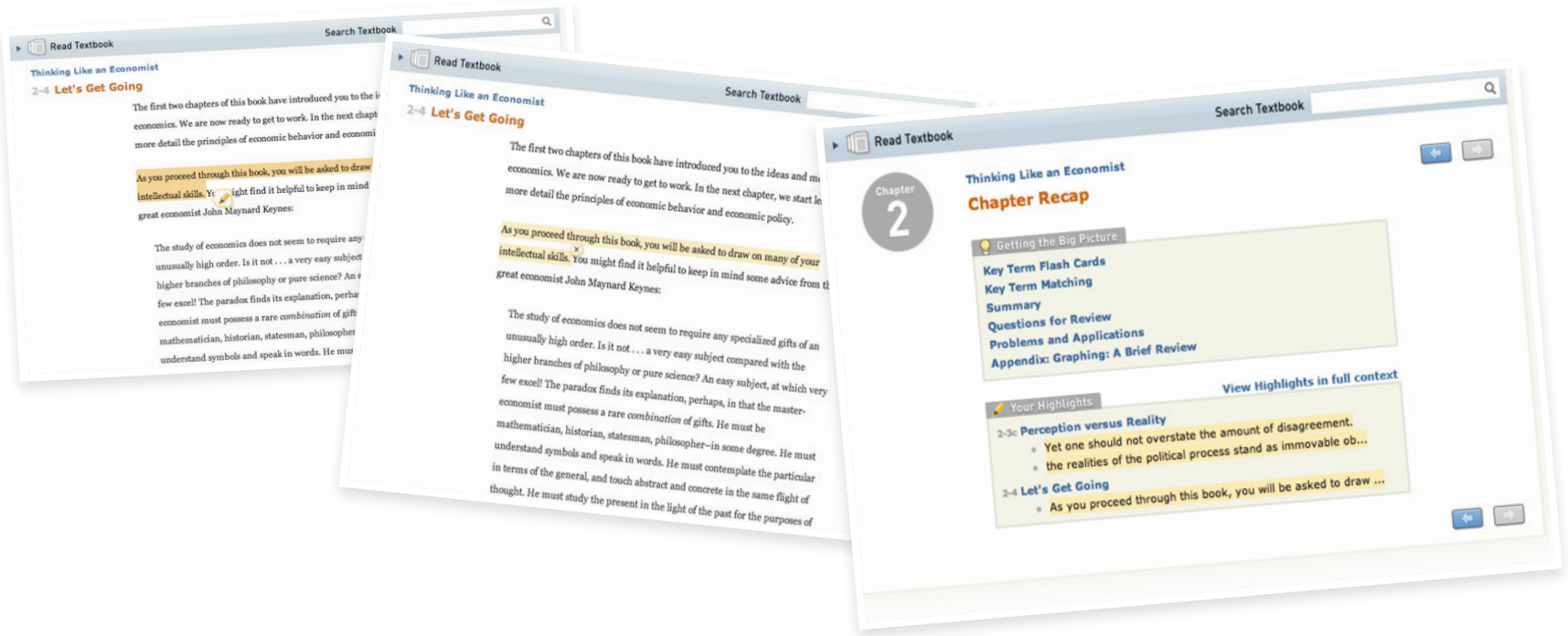
Aplia Text: Process



Next, I started playing around with using rollover to reveal more information about a given page (such as the first paragraph). After discussing it with one of our developers, we came up with a way to use the underlying XML to semantically extract key phrases from a given page and display them as a summary.

Aesthetically, you'll see that the design continued to evolve and simplify.

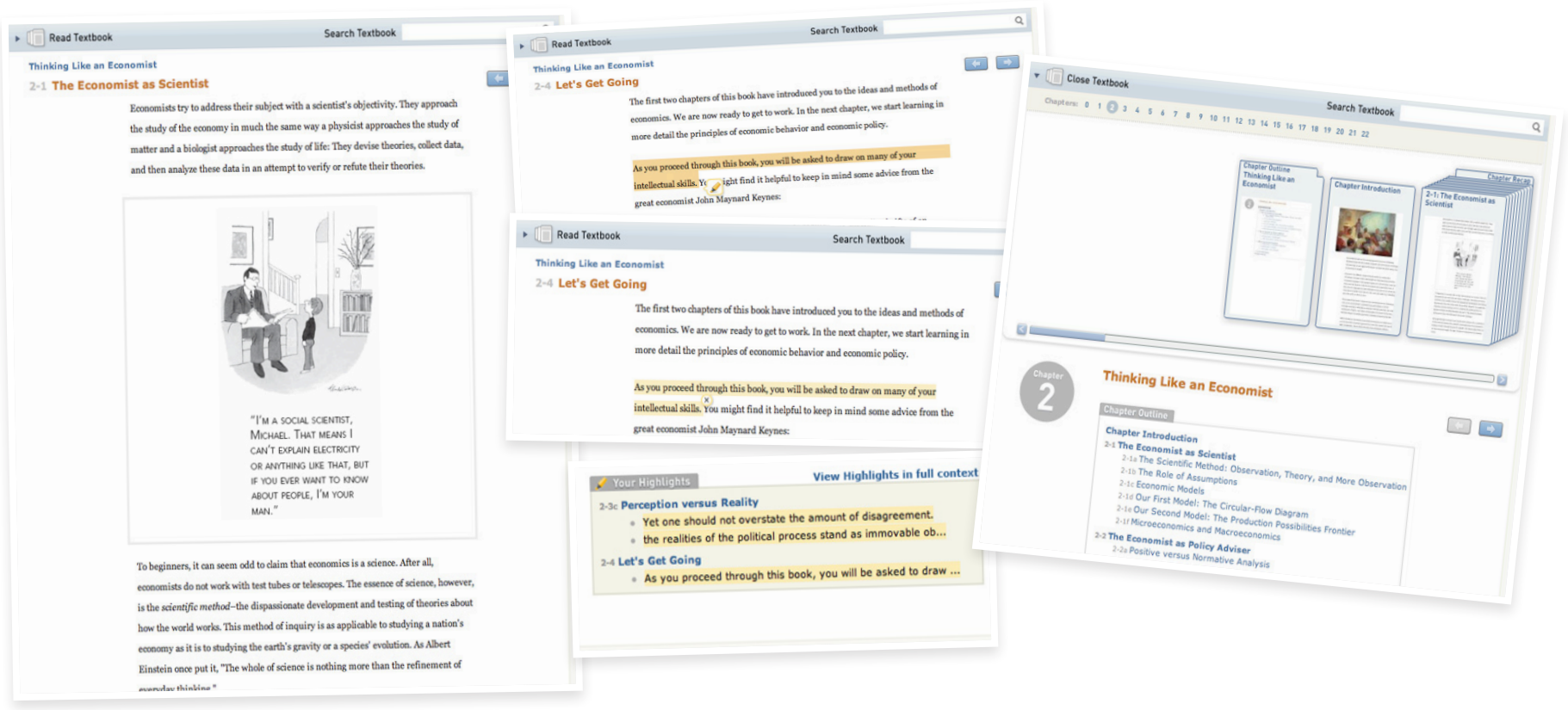
Aplia Text: Process



As I had mentioned, one of the primary things we saw students do in their physical books was highlight. However, when we looked at usage on other ebook platforms, we saw very few students using their highlight features. After talking to students and trying it out ourselves, it became obvious to me that the problem was a UX one. It simply was a pain to highlight in these other platforms.

Inspired by radial menus from video games, I came up with the concept of displaying the option to highlight once students had selected text. Interestingly, about a year later, Apple released an update to iOS that used a similar concept for copy/paste. We also aggregated all of a given chapters' highlights into one place, essentially enabling students to build their own study guide.

Aplia Text: Process



Final screenshots.

Demo video: http://www.youtube.com/watch?v=RYVty91hhIY

Live demo (follow link and click "View a chapter of the digital textbook"): http://aplia.com/demo?id=124