

Call me Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. **Mobile Web Annotation.**  

is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth, whenever it is a damp, drizzly November in my soul,

Ian McDowell; Alison Meier; Jessica Voytek;

Mobile Web Annotation

Alison Meier, Ian McDowell, Jessica Voytek

University of California at Berkeley
INFO 213 "User Interface Design and Development"
Spring 2010

Contents

Names and Roles	7
Process Map	8
Problem and Solution Overview	10
Tasks	10
Design Evolution	11
Brainstorm	11
Contextual Inquiry	12
Paper Prototype	16
Paper Prototype Testing	18
Interactive Prototype	20
User Study	22
Element Evolution	24
Main Page Buttons	24
Linked Notes	24
Annotation & Review Mode	25
Method of Highlighting	25
Help Animation	25
Returning to the Original Text from Review page	25
Most Valuable Technique	26
Final Interface Design	26
Overview	26
Features	26
Technical Implementation	28
Limitations and Features Not Implemented	31



Names and Roles

The philosophy of our development process borrowed heavily from principles of agile development. We emphasized co-location for our working sessions and accomplished the vast majority of coursework while sitting together in the same room. This allowed us to collaboratively find solutions to problems as they arose, and to make rapid shifts in design and implementation choices when necessary.

All group members participated in the design and execution of our interface throughout the various stages of development. We were all present and active in running the various interface studies we conducted and worked through assignment write-ups as a team. Beyond the shared work, members of the team had some specialized duties based on their strengths, as listed below:

Jessica Voytek (*Scout Leader*): Jessica was the lead graphic artist and generated all icons and graphics used in our interface. She was also a major force in front-end development, and coded the bulk of the JavaScript functions used in the high fidelity prototype, along with much of the HTML/CSS.

Alison Meier (*Editor in Chief*): Alison took the lead on write-ups by contributing a disproportionate amount of text for the assignments and using her linguistic prowess to improve the written content created by the rest of the team.

Ian McDowell (*Grand Poobah*): Ian was lead designer on the back-end of the high fidelity prototype, coding the lion's share of the server-side scripts. He also contributed to the front-end logic and code, handling most of the 'review' page implementation.

Brainstorm

"Wouldn't it be nice if we could annotate anywhere?"

Contextual Inquiry

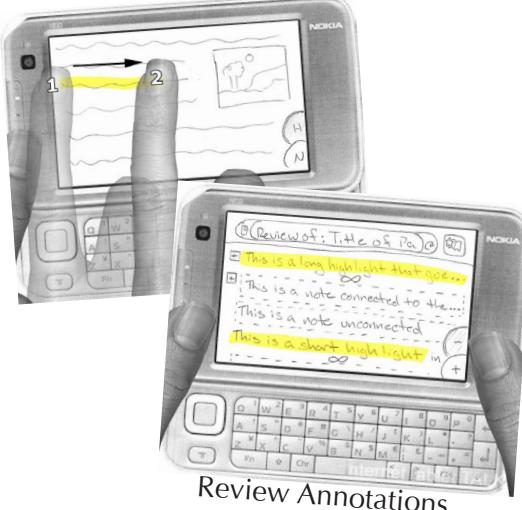
Target Users

- Read > 3 academic papers per month
- smartphone owners
- diverse needs

Inquiry

Initial Design Sketches

Annotate



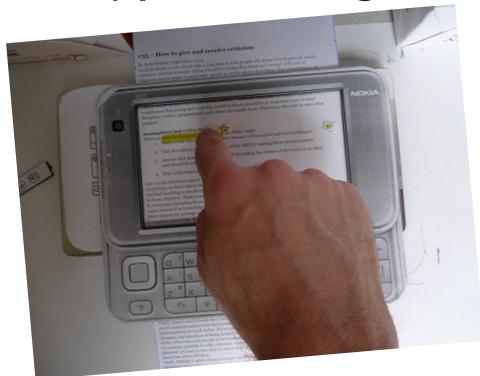
Review Annotations

Process Map

Participant Behaviors

- highlight full sentences & single words
- make notes in margins
- mark some things important
- review notes

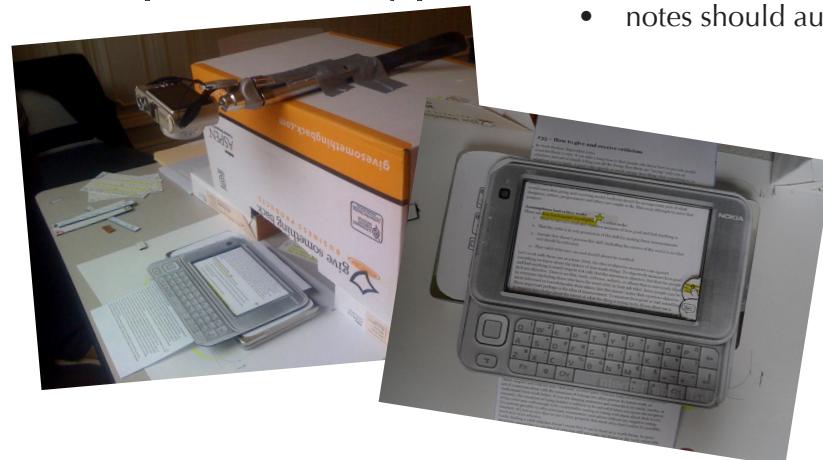
Paper Prototype Testing



Findings

- finger-drag highlight = intuitive
- note/highlight buttons obvious
- review button = not obvious
- return to page = not obvious
- notes should auto-link w/highlights

Paper Prototype



Interactive Prototype

The Poetics: Aristotle on the Art of Poetry

II. The objects the imitator represents are actions, with agents who are necessarily either good men or bad—the diversities of human character being nearly always derivative from this primary distinction, since the line between virtue and vice is one dividing the whole of mankind. It follows, therefore, that the agents represented must be either above our own level of goodness, or beneath it, or just such as we are in the same way as, with the painters, the personages of Polygnotus are better than we are, those of Paion—worse, and those of Euphranor just like ourselves. It is clear that each of the above-mentioned arts will admit of these differences, and that it will

Mobile Web Annotation Mock-Up : "The Poetics"

The Poetics: Aristotle on the Art of Poetry

II. The objects the imitator represents are actions, with agents who are necessarily either good men or bad—the diversities of human character being nearly always derivative from this primary distinction, since the line between virtue and vice is one dividing the whole of mankind. It follows, therefore, that the agents represented must be either above our own level of goodness, or beneath it, or just such as we are in the same way as, with the painters, the personages of Polygnotus are better than we are, those of Paion—worse, and those of Euphranor just like ourselves. It is clear that each of the above-mentioned arts will admit of these differences, and that it will

Mobile Web Annotation Mock-Up : "The Poetics"

http://annoapp.appspot.com/review/

Technical Limitations

- no mouse events--changed to single/double tap for highlight
- created page-based mock-up rather than browser extension

Design Improvements

- moved review button to main screen
- changed icons to letters due to size
- link notes automatically

Heuristic Evaluation

Findings

- not clear highlight mode is on
- slow double-click = cumbersome
- single-letter buttons difficult to understand
- separation between highlights unclear

Modifications

- removed highlight button
- tap 1st and last word to highlight
- display help animation
- changed button abbreviations to full words
- added lines between highlights on review page

Usability Study

Findings

- highlight was unintuitive
- users dismissed help animation before seeing it
- help animation was ambiguous
- still unclear how to return to original text from review page

To Highlight...

A short message is being shown. Tap anywhere to dismiss this message.

Mobile Web Annotation Mock-Up : "The Poetics"

A sharp increase in nonresident admissions will bolster campus efforts to raise revenue, but some students question whether such a move will jeopardize the campus's public character.

The campus admitted 13 percent more nonresident students for the upcoming fall 2010 semester compared with last year. Other UC campuses recorded decreases in resident admissions that were less dramatic, according to data released by the UC Office of the President. According to UC Berkeley officials, the increase in nonresident admissions will generate more money for the campus to reconcile "insufficient state funding for California student enrollment."

This is a note.

Admissions will generate more money for the...

Return to Text

Mobile Web Annotation Mock-Up : "The Poetics"

A sharp increase in nonresident admissions will bolster campus efforts to raise revenue, but some students question whether such a move will jeopardize the campus's public character.

The campus admitted 13 percent more nonresident students for the upcoming fall 2010 semester compared with last year. Other UC campuses recorded decreases in resident admissions that were less dramatic, according to data released by the UC Office of the President. According to UC Berkeley officials, the increase in nonresident admissions will generate more money for the campus to reconcile "insufficient state funding for California student enrollment."

This is a note.

Admissions will generate more money for the...

Return to Text

Final Interface

Modifications

- added (?) button to recall help animation
- modified help animation with "sound" effects
- added textual cues to animation
- added link "Return to Text" on review page

To Highlight...

A short message is being shown. Click the beginning and end of a phrase to highlight. Tap anywhere to dismiss this message.

Mobile Web Annotation Mock-Up : "The Poetics"

A sharp increase in nonresident admissions will bolster campus efforts to raise revenue, but some students question whether such a move will jeopardize the campus's public character.

The campus admitted 13 percent more nonresident students for the upcoming fall 2010 semester compared with last year. Other UC campuses recorded decreases in resident admissions that were less dramatic, according to data released by the UC Office of the President. According to UC Berkeley officials, the increase in nonresident admissions will generate more money for the campus to reconcile "insufficient state funding for California student enrollment."

This is a note.

Admissions will generate more money for the...

Return to Text

Mobile Web Annotation Mock-Up : "The Poetics"

A sharp increase in nonresident admissions will bolster campus efforts to raise revenue, but some students question whether such a move will jeopardize the campus's public character.

The campus admitted 13 percent more nonresident students for the upcoming fall 2010 semester compared with last year. Other UC campuses recorded decreases in resident admissions that were less dramatic, according to data released by the UC Office of the President. According to UC Berkeley officials, the increase in nonresident admissions will generate more money for the campus to reconcile "insufficient state funding for California student enrollment."

This is a note.

Admissions will generate more money for the...

Return to Text

Design Evolution

Brainstorm

In our original brainstorming session, we came up with the idea of wanting to be able to create annotations anywhere. We wanted to add notes and highlights to web pages on a mobile device. We thought we could create a summary of the highlights and store it in the cloud. We also wanted to capture automatic metadata about notes using the phone's capabilities, such as recording the date, time, and GPS location where the note was created. We thought that adding tags to notes would be useful. We also thought that it would be useful to be able to share annotations with others through something like an RSS feed.

From our brainstorming session, we developed a project plan to create a Firefox Mobile browser extension with a more advanced desktop component. We wanted the notes to be able to sync across various devices. We anticipated that users would want the ability to annotate material while on their mobile devices but do more complex functions on a regular computer. We wanted users to be able to access a summary report on both the mobile device and the computer version, with the computer version having more complex features such as a search function and access to automatic metadata and tagging.

- A way to annotate the Web
- Adding highlights and notes to web pages
- Display highlights as a summary and save
- Store notes in the cloud
- Browse annotation without visiting site
- Use automatic capture to get GPS, date, time when note was taken
- Connect to calendar for info about where you were when note was taken
- Be able to add tagging (auto and custom) to notes
- Share annotations like RSS for important browsers (or individuals)

- 20) Annotate Func
* highlights
* Notes
- 21) Transfer Note
- 22) Display Highl
- 23) Store Notes
- 24) Browse Annotation Site
- 25) Camera / Phone ^{Audi}
- 26) Automatiz GPS Da
- 27) Connect Cal
- 28)

Problem and Solution Overview

Our goal was to allow users to interact with academic text in a meaningful way at any location. Our solution is a lightweight browser-based annotation application that runs on a mobile handset. We use cloud-based storage to allow highlights and notes to be recorded anywhere on any device, along with review functionality that allows users to quickly skim their annotations. The system was designed with the intention that users could switch to a full-featured desktop annotation management application for tasks that benefit from richer interaction.

Tasks

Our final task list was as follows:

1. Read through the text. Highlight a few words and sentences that you find interesting.
2. Mark one of the highlights as important.
3. Create a note that says, "This is important."
4. Edit the note to say, "This is really important."
5. Enter review mode.
6. Expand and contract one highlight on the page.
7. Expand all notes and highlights on the page
8. See a note in its original context.

While our task list stayed thematically consistent over the course of the project, adjustments were made as the interface evolved and some tasks changed in importance or became irrelevant. Tasks (1), (3), and (5) were present from the beginning, and task (4) is little more than an extension of task (3). Task (2) was added along with the starring feature, based on user desire to designate some annotations as having special significance. Tasks (6)-(8) were added with the low-fidelity prototype, which fleshed out the review page and necessitated testing of the new features.

These new tasks replaced others that were dropped through the course of development. Our very first task list contained the task of manually linking a note to an associated highlight. User testing of the paper prototype showed that most users expected for this to happen automatically, so we changed the functionality and removed the task.

We had also originally envisioned tasks corresponding to a desktop application capable of performing more advanced search and editing on highlights. For scoping reasons we decided to focus on the mobile application, and thus deleted such tasks.

Lastly, one of our long-standing tasks called for the user to "turn on" annotation mode. This operation made sense when we were envisioning implementing our application as a Firefox plugin. For technical reasons, our final prototype was instead created as a proof-of-concept stand-alone web application, where an intention to annotate was assumed. Thus, the annotation feature is always "on," and there is no need to enable it.

Design Evolution (cont.)

We surveyed existing web annotation systems. We compared our plan to several existing annotation systems--Web Notes, everNote, uberNote, and clipMarks. None of these existing systems allowed users to highlight web pages on a mobile device.

Product	Highlight Web Pages	Mobile Version	Organize Notes	Sync w/ Desktop	Light-weight	Simple Mobile Interface	Location/Time Metadata
Web Notes	x		x				
everNote		x	x	x			x
uberNote		x	x	x			
clipMarks		x			x	x	
Our Project	x	x	x	x	x	x	x

Contextual Inquiry

Target Users

We decided that our target users are higher education students who are already smartphone owners and who currently read at least three academic papers per month. For our contextual inquiry, we wanted to observe target users who fit these criteria. We sought participants from a variety of academic disciplines who studied in a variety of environments. We specifically sought at least one participant who studied on public transit, since studying on-the-go is an ideal use case for our system.

Environment

We wanted to observe students reading academic materials in their natural studying environment. We approached subjects and asked them to let us observe as they worked, as well as to interrupt and ask questions as needed. We recorded information such as the type of study materials that the participants used (e.g. printed papers, books, files on a laptop) and what tools they used to annotate (e.g. highlighters, pens). We paid close attention to how each participant annotated material. We asked for details about why they highlighted or notated particular portions of the text. We observed how

much text they highlighted, whether they highlighted full sentences or single words, when they added text in the margin, and whether they added any other special symbols or notes. We also asked the participants how they use their notes later and whether they had any special systems for review.

"[I make a note] if it was a major point. I highlight too much to find it without writing in the margins"

Design Evolution (cont.)

Contextual Inquiry Participants

No.	Academics	Location	Study Materials	Key Findings
1	Third-year PhD candidate in neuroscience	Coffee shop	Printed documents or PDFs on a laptop	<ul style="list-style-type: none">• Usually read papers to determine whether they were relevant to her research• Made notes and highlights on things other than text• Sometimes drew pictures to clarify her point• Read abstract first, then figures, then “methods” (how the experiment was conducted), then rest of text
2	First-year law student	BART train	Textbook	<ul style="list-style-type: none">• Switched between a pen and a highlight frequently• Highlighted large sections of text and summarized section in the margin• Also highlighted single words• Physical book was often in the way, had to move out of the way for other passengers• Did not like using her laptop on BART due to concern about safety
3	Master's student in city planning	Common area of a large house	Printed documents	<ul style="list-style-type: none">• Put off reviewing until the last minute• Put a check at the top of an article to verify having read the paper• Frequent social interruptions while reading
4	Recent undergraduate in Peace and Conflict studies	Home, in bedroom	Printed documents	<ul style="list-style-type: none">• Liked being able to see the whole document to see relative density of highlights• When reviewing, would transcribe significant highlights into Microsoft Word in order to create a review sheet• Starred key materials

Design Evolution (cont.)

Interface Sketches

We designed a mobile annotation system that works as a Mobile Firefox extension. We designed our system with the Nokia N900 in mind.

Our extension adds two buttons to the Mobile Firefox right menu bar, which one accesses by swiping to the left. One button is the annotate button, which the user can select to turn the extension on (Figure 1).

Once the extension is turned on, highlight and note buttons appear in the bottom right corner. We chose the bottom right corner for convenient thumb access.

When the user selects the H (highlight) or the N (note) button, a button bubbles out beneath it so that the user can be sure he or she is selecting the desired button (Figure 2).

When the highlight button is selected, the user has turned on highlight mode. This prevents the screen from sliding left or right to the menu bars.

The user highlights by selecting and dragging a finger across selected

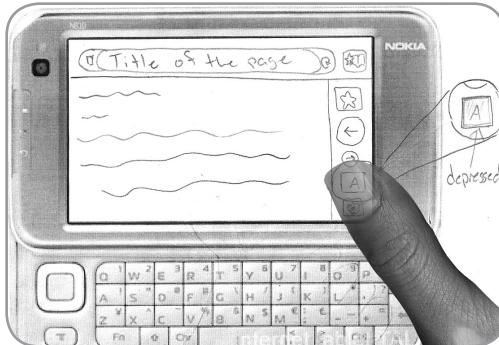


Figure 1

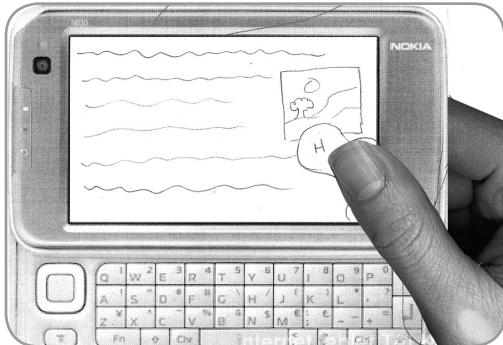


Figure 2

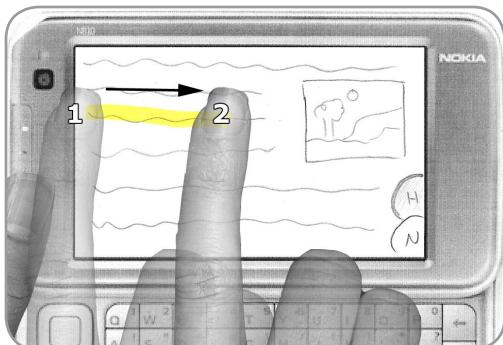


Figure 3

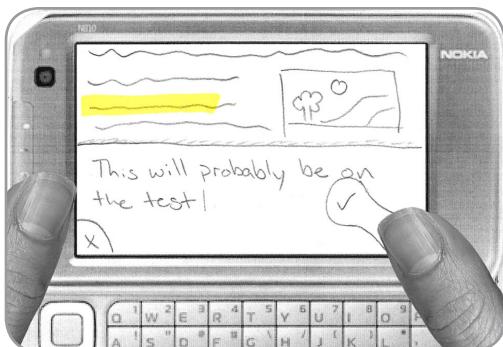


Figure 4

Design Evolution (cont.)

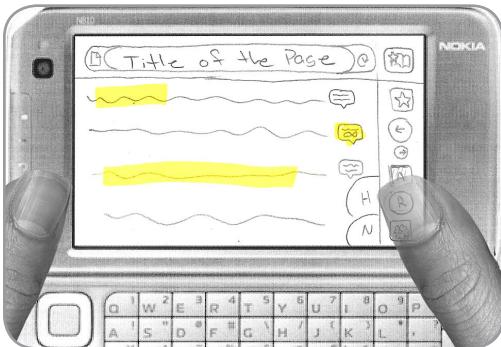


Figure 5

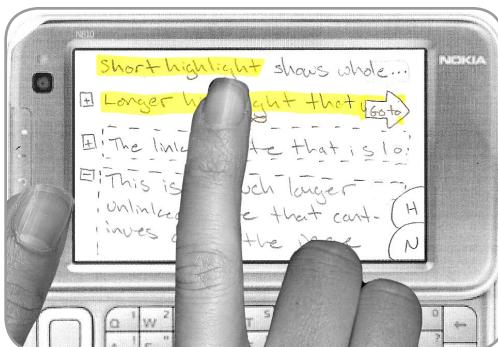


Figure 8

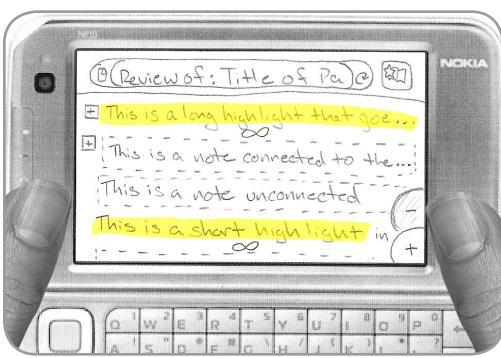


Figure 6

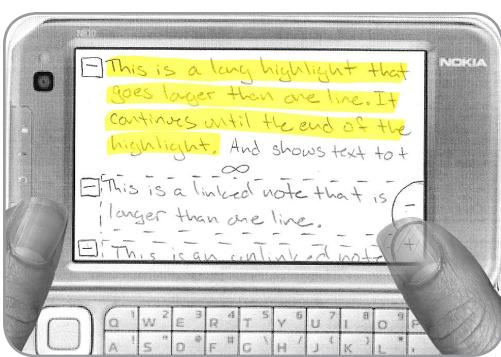


Figure 7

A user can review the highlights and notes made on the page by sliding back to the right menu bar and selecting the review button (Figure 5).

By default, all of the highlights are one line in length so that the user can get a quick view of the highlighted material. Truncated notes are followed by ellipses, and the user can expand or contract individual notes. If a user has highlighted a single word or a short amount of text, text surrounding the highlight also appears. This helps give short highlights context (Figure 6).

The user can also use the + and - buttons under the right thumb to expand and contract all of the notes on the review page (Figure 7).

The user can return to the original page from the review page by tapping a highlight or note once. A "go to" arrow then appears, and the user can tap this arrow to return to the original page (Figure 8).

Design Evolution (cont.)

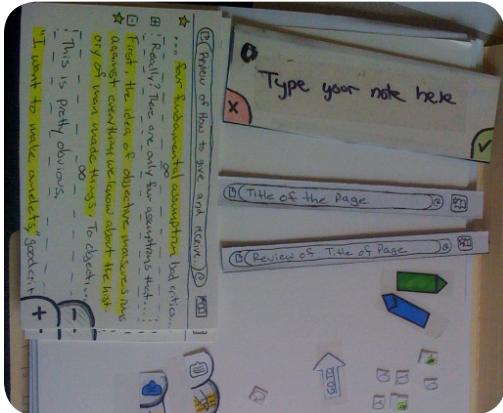


Figure 9



Figure 10

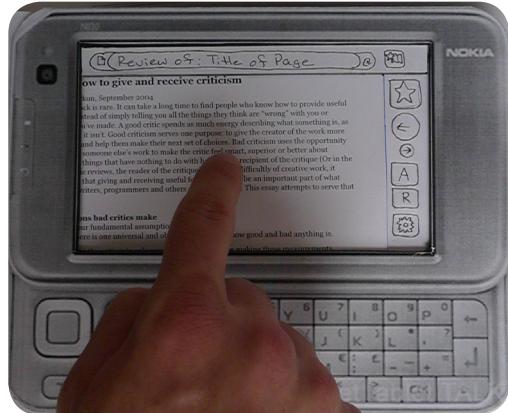


Figure 11

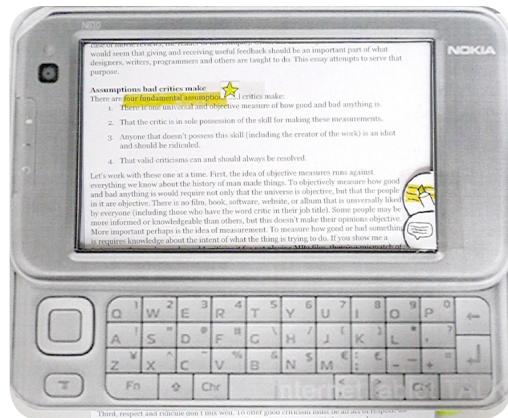


Figure 12

Paper Prototype

Paper Prototype Design

Creation

We created a paper prototype of our design out of paper and cardboard materials with magnetic backing (Figure 9). We used a magnetic metal box as a base in order to keep all of the small pieces in place.

For the frame, we enlarged a Nokia phone image and backed it with cardboard. We cut out the screen, then attached magnets to the back so that it would stick to the metal base. The magnets also served as guides

that allowed the user to mechanically “scroll” a paper web page inserted between the cardboard overlay and the box. For on-screen elements, we used paper backed with removable sticky materials.

Video

We also created a video demonstrating the paper prototype using quasi-stop motion techniques. We used burst mode on a camera to shoot successive photos and then edited the photos together to create a video (Figure 10).

Design Evolution (cont.)

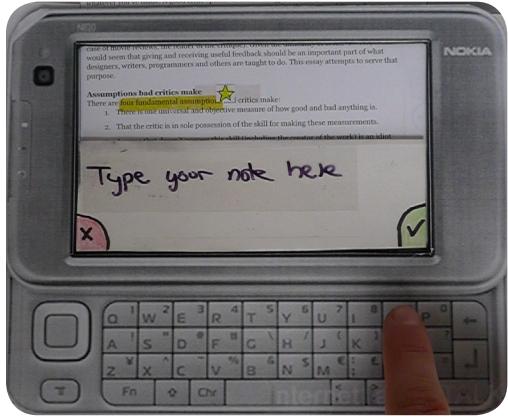


Figure 13

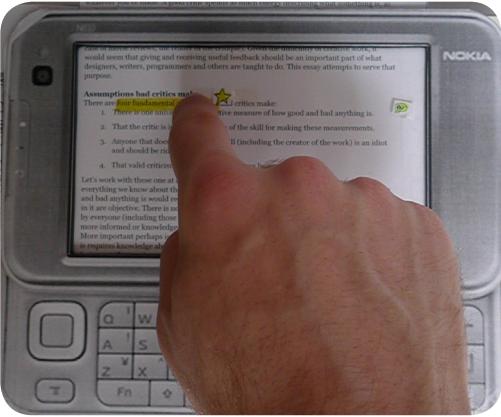


Figure 14

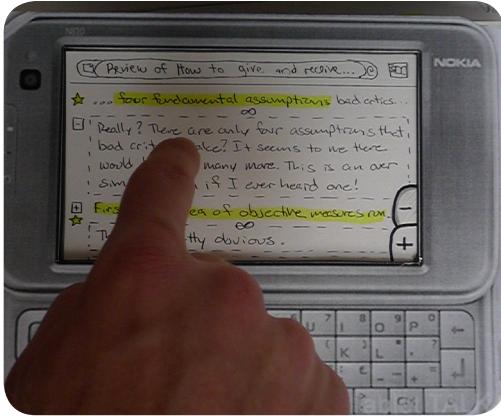


Figure 15

Functionality

Using the paper prototype, a user could scroll to the left and right to access the Mobile Firefox menu bars. From there, the user could select “A” for annotation mode. The highlight and note buttons would then appear on the main screen. While creating the paper prototype, we changed the “H” and “N” abbreviations on the buttons to icons because we thought users would find icons easier to understand (Figure 11).

From the main screen, the user could select the highlight button. The user could then use a finger-drag motion to highlight text (Figure 12). In the paper prototype, we added the option for

users to star important material. A star outline appears following the text after the user makes a highlight. If the user taps the star, the outline fills in and the star remains. If the user does not tap the star, it disappears when the user taps elsewhere. Highlight mode turns off after the user highlights something.

The user can also select the note button from the main screen. A text box fills half the screen (Figure 13). The user can type text into the text box. If the user selects the red X, the note is deleted. If the user selects the green check in the bottom right, a note is created. A note icon is added to the middle of the screen. The user can connect the note to a highlight by tapping the note and

then the highlight (Figure 14). The note icon turns yellow and moves to the margin of that highlight.

The user can return to the right-hand Mobile Firefox menu and select “R” for review mode. This loads another page that displays a summary view of the highlights, notes, and stars (Figure 15). The default view of review mode is that long highlights are shortened to one line. The user can select + or - buttons to the left of each annotation to expand or contract individual items, or the user can use the + and - buttons under the bottom right thumb to expand or contract globally.

Design Evolution (cont.)

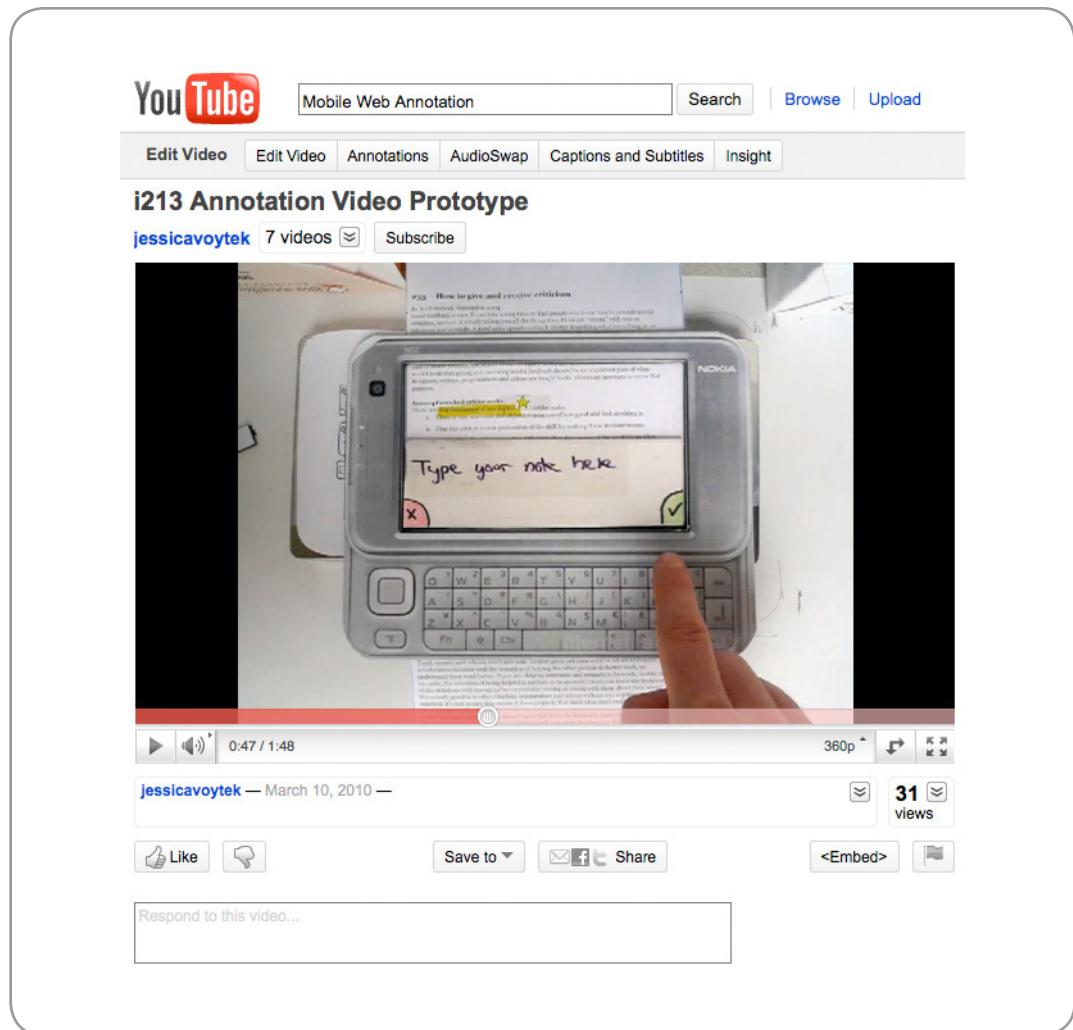
Paper Prototype Testing

We tested our paper prototype with three participants. One of us acted as the facilitator, one as the “computer” who moved around the paper prototype pieces, and one as the note taker.

After introducing ourselves, asking participants to sign a consent form, and giving a basic demonstration of the prototype, the facilitator asked each participant to complete a series of tasks. As the participant completed one task, another task was given.

Results

As a result of our paper prototype tests, we learned that users found the finger-drag highlighting method very intuitive. The highlight and note buttons on the main page were clear and easy to use. On the other hand, we also learned that the review button was hard to find in the browser menu. Also, users assumed that notes would automatically link to the previous highlight. The participants also had difficulty returning to the main page from the review page.



See the paper prototype video <http://bit.ly/d1ZxRu>

Design Evolution (cont.)

Tasks

We asked each participant to complete the following tasks:

- 1) Easy - Turn on annotation mode, highlight text, and mark it as important.
 - a) Turn on annotation mode.
 - b) Highlight some text.
 - c) Mark the highlight as important.
- 2) Medium - Create note and link it to the highlight.
 - a) Create a note reading: "This is important."
 - b) Link the note to highlight.
- 3) Medium - Turn on review mode, determine what items are notes/ highlights, and how they're linked.
 - a) Enter Review Mode

- b) Locate the highlights on the page
- c) Locate the notes on the page
- d) Determine which notes are linked to which highlights
- 4) Medium - Use the different expand and contract functions to review long notes.
 - a) Expand and contract one long note or highlight on the page
 - b) Expand all notes and highlights on the page
- 5) Hard - See a note in its original context.

Measurements

We logged critical incidents that we observed, assigned severity ratings, recorded the number of times help was requested, evaluated task completion, and counted the number of errors each participant made. Below is the log of the critical incidents we observed. We independently rated each problem then averaged the three scores to get the final rating. Severity ratings were based on the following scale:

- 1) I don't agree this is a usability problem.
- 2) Cosmetic problem
- 3) Minor usability problem
- 4) Major usability problem
- 5) Usability catastrophe

Issue	Severity Rating
User unable to tell difference between highlight mode on and highlight mode off	2
Notes don't automatically connect to the last highlight	3.66
Difficult to find the link icon between linked notes and highlights in review mode	3.66
Difficult to find the "Go To" link in review mode	2.66
Not clear what the "A" and "R" buttons do	2
Trying to highlight before turning on review mode	1.66
Highlight mode turns off after making a highlight	1.66
Unclear to users that option to expand/contract notes is available	2.33

Design Evolution (cont.)

Interactive Prototype

Mockup

Our original plan was to create a Mobile Firefox extension. We attempted to do this as our interactive prototype. Since Mobile Firefox is so new, there is very little documentation or user guides for developing content. We eventually abandoned the plan to create an extension for the purposes of the prototype. Instead, we created an HTML/JavaScript/CSS mockup with a Google app engine back-end.

Technical Changes

Even though we created a mockup instead of a true extension, we ran into several technical limitations that required us to change some design elements. Mobile Firefox does not support mouse events, so we were unable to create some of the functionality that we desired. Because our interactive prototype was not a true extension, we no longer had access to the browser menus. We eliminated the “annotation on” button for the purposes of our prototype. Although we wanted to use finger-drag method

of highlighting, the touch screen functions of sliding left and right to access toolbars override the traditional desktop option of selecting text. Instead of selecting the text to be highlighted by touching and dragging a finger, we changed this functionality to the user taps once to select the current word, twice to select the entire sentence. Because double tap is a built-in feature in Mobile Firefox that zooms in on the screen, the user has to do a slow, deliberate double tap to highlight.

In addition, we were not able to use a mouse-down functionality as we had planned, so we were no longer able to show the note and highlight buttons “bubbling out” beneath the user’s finger.

Design Changes

We also made some design changes, based on our findings from the paper prototypes testing and our own ideas. We moved the review button to the main page rather than the browser menu. We also labeled the review buttons with “R,” “H,” and “N” rather than using icons. Though we preferred the icons, the buttons were too large and covered up text. We changed

the button design to be as narrow as possible (though the field to select the buttons is larger than the buttons themselves) so that they do not overlap text on the page (Figure 16).

In order to address findings from the paper prototype testing, we changed the notes to link automatically to the last highlight. In review mode, we eliminated the “Go To” arrow. Users are able to return to the original text with a single tap on the annotation (Figure 17).

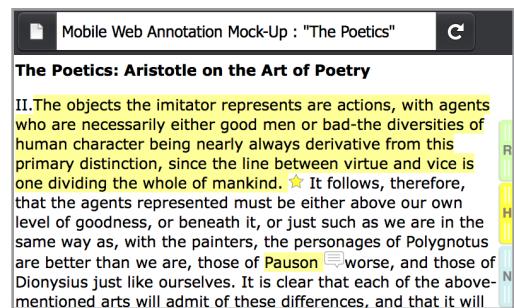


Figure 16

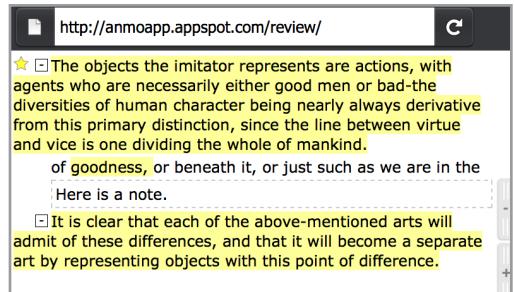


Figure 17

Design Evolution (cont.)

Heuristic Evaluation

Findings

In order to evaluate our interactive prototype, we performed a heuristic evaluation with one participant using a Mobile Firefox desktop simulator. We found that our participant had trouble seeing when highlight mode was selected. She also found the slow double click cumbersome. She didn't understand what the button abbreviations "R," "H," and "N" meant.

She also had difficulty distinguishing between the individual highlight and notes on the review page.

Modifications

Based on the heuristic evaluation feedback, we made additional changes to the interactive prototype. We realized that we no longer needed a highlight mode at all, so we eliminated the highlight button entirely. Instead of the single tap (single word) and slow double tap (full sentence) highlight

method, we changed the highlight method to tap the first word and then the last word of the desired highlight section. Since this is an unusual method for highlighting, we created an animated help splash screen that appeared when the user first loaded the page.

We also changed the button labels to spell out "note" and "review" so that they would be clearer. Furthermore, we added a horizontal rule between highlights on the review page.

#	Heuristic	Location	Description	Sev.	Fix.	Sum	Possible Fix
1	feedback	main page buttons	did highlight button change? "pretty subtle"	3	3	6	Remove Highlight Button - we don't need it
2	prevent errors	main page	wants to click and drag or double click	3	2	5	User education splash screen
3	consistency	following highlights	don't understand why some have stars and some don't	1	2	3	Documentation
4	simple and natural dialog	following highlights	can't return to a highlight and star it later	2	2	4	Implement ability to tap highlight and revisit star
5	speak the user's language	main page buttons	difficult to understand what R, H, and N mean	3	3	6	change buttons to spell out full words
6	speak the user's language	review page	not clear that review mode highlights are separate snippets	1	4	5	horizontal rule between highlights
7	feedback	review page	plus and minus buttons don't do anything when there is nothing to expand or contract	2	2	4	could grey out or make buttons unavailable

Design Evolution (cont.)

User Study

Method

We tested our revised interactive prototype with three more participants. The participants were all undergraduate or graduate students. This is the first time that we tested the prototype using the Nokia N900. One of us acted as the facilitator, one as the note taker, and one as a videographer to record the tests.

We introduced ourselves and explained the study, asked participants to sign a consent form, and gave a brief overview of how the phone and the browser worked. We then gave them a printed list of tasks that we asked them to complete. We debriefed with the participants following the task completion.

Tasks

We asked them to complete the following tasks:

- 1) Read through the text. Highlight a few words and sentences that you find interesting.
- 2) Mark one of the highlights as important.

- 3) Create a note that says "This is important."
- 4) Edit the note to say "This is really important."
- 5) Enter review mode.
- 6) Expand and contract one highlight on the page.
- 7) Expand all notes and highlights on the page
- 8) See a note in its original context.



Design Evolution (cont.)

Measurements

We recorded the time each participant took to complete each task and the number of times that each needed assistance or asked questions. We also logged critical incidents.

Results

By far the largest problem we encountered was with the user understanding of how to initiate a highlight. Users were strongly inclined to use a swiping gesture to highlight, as they would with a real pen. We

had hoped that the help screen would inform users that this was not the way to highlight, but this was unsuccessful because users dismissed the help page without reading it or did not understand it. In addition, users struggled to figure out how to return to the original text by single tapping on the annotation.

We did find that users were successfully able to add and edit notes, star highlights, review annotations, and expand and contract annotations in review mode.

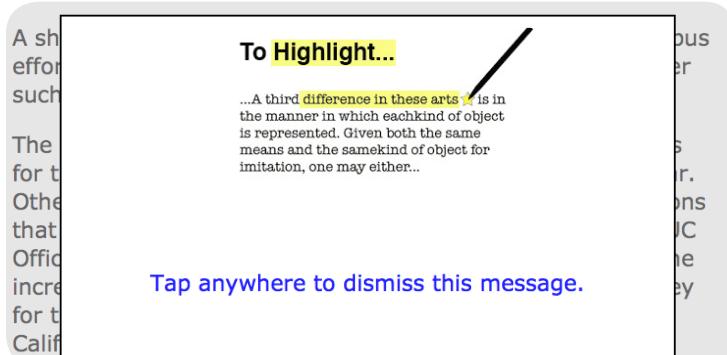


Figure 18 - Old Help Screen

Modifications

In order to address the issues from the user test, we clarified the animation on the help screen (Figures 18 & 19). We also added a button to allow the user to recall the help screen from the main page. Finally, we added a link on the review page to return to the original text. The final design will be discussed in greater detail in the design section.

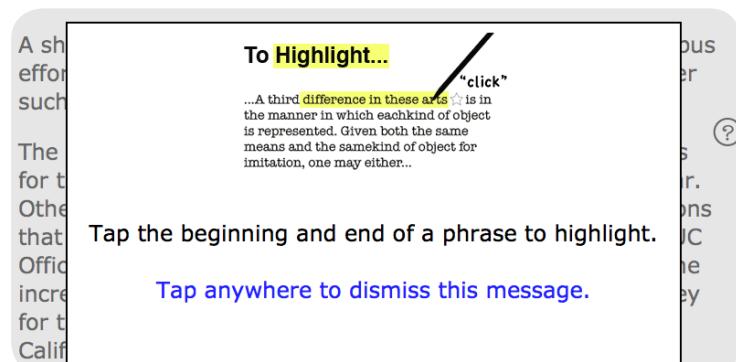


Figure 19 - New Help Screen

Element Evolution

Main page buttons

Our initial sketch showed an "H" and an "N" button in the bottom-right corner with a "bubble-out" function. In the paper prototype, we changed these to depict a highlighter and note icon. In the interactive prototype, we found that the icons covered up text, so we made the buttons as narrow as possible and changed the labels to single-letter abbreviations. We also eliminated the "bubble-out" function because the system does not support mouse-down. In the final version, we removed the highlight button entirely because we realized it was no longer necessary. We also moved the review button to the main page and spelled out the full words. We also added a help button to recall the help animation. The note and review buttons are not available until the user has created a highlight.

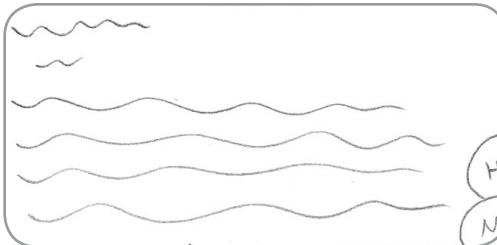


Figure 20 - Contextual Inquiry Sketch

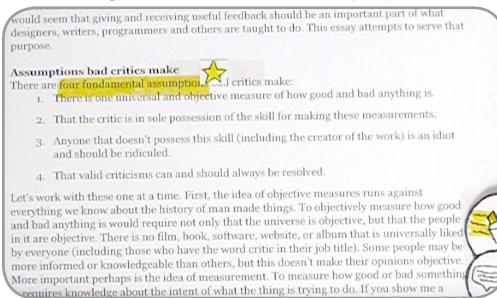


Figure 21 - Paper Prototype

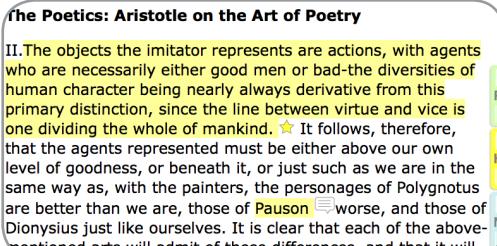


Figure 22 - Interactive Prototype

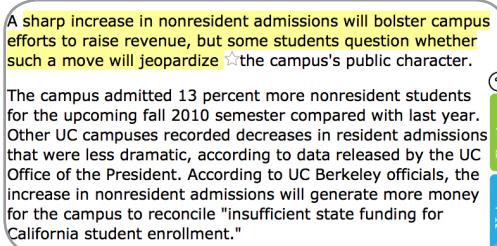


Figure 23 - Final Prototype

Linked notes

In our initial sketches and paper prototype, we gave users the option to connect a note to a highlight. Our paper prototype participants told us that they thought notes should be automatically linked to the last highlight. In the interactive prototype, notes link automatically to the last highlight.

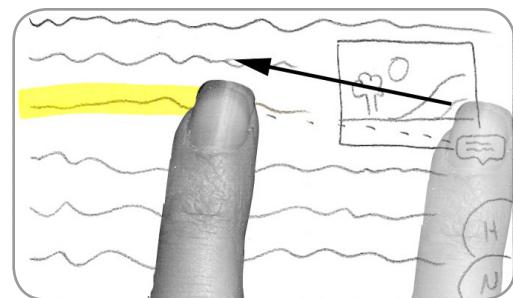


Figure 24 - Contextual Inquiry Sketch

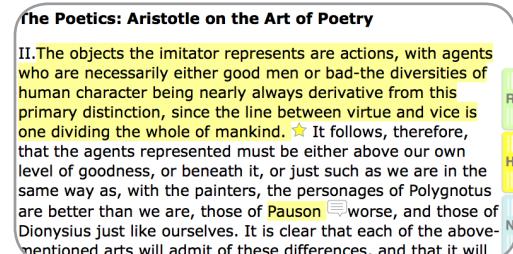


Figure 25 - Interactive Prototype

Element Evolution (cont.)

Annotation and review mode

In our initial sketches and paper prototype, we planned to have users access the right-hand Mobile Firefox menu to turn on annotation mode and move to review mode. Since we did not end up making an extension, annotation mode is always "on" in the prototype. The review button was moved to the main page.

Method of highlighting

We planned to use a tap and drag system of highlighting. When we created the interactive prototype, however, we changed this to a single tap (single word), slow double tap (full sentence) method to accommodate system limitations. In our heuristic evaluation, we found the method cumbersome, so we changed to a tap at the beginning and end of the text.

Help animation

We added help animation before the usability test to explain the new method of highlighting. During the usability test, we realized that the animation was not clear and revised it.

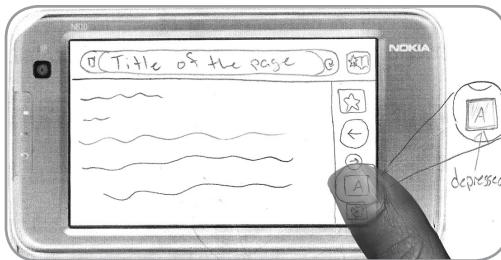


Figure 26 - Contextual Inquiry Sketch

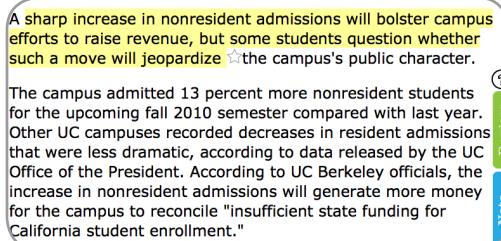


Figure 27 - Final Prototype

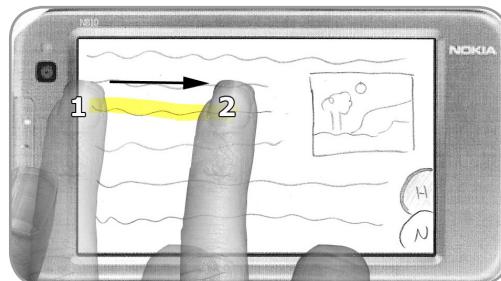


Figure 28 - Contextual Inquiry Sketch

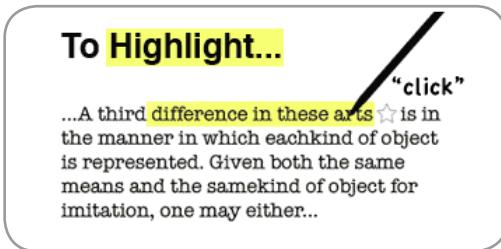


Figure 29 - Final Prototype

Returning to original text from review page

We originally planned for the user to return to the original text from the review page by tapping once on the annotation. A "Go To" arrow would appear, and then users could tap on that. We found that users thought the "Go To" arrow was unnecessary, so we changed this in the interactive prototype to return after a single tap. In the usability study, the participants found this unclear, so we also added a "return to text" link in the final version.



Figure 30 - Contextual Inquiry Sketch

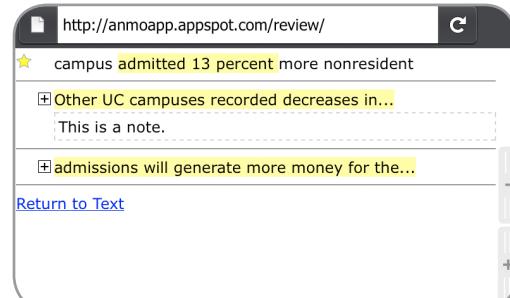


Figure 31 - Contextual Inquiry Sketch

Most Valuable Technique

Our system evolved during each stage of the process. The contextual inquiry gave us direction about how people take notes while they study. The sketches helped us think through our ideas and create a common vision for the system. Testing the paper prototype gave us the chance to show our implementation to others to see if our ideas met users' ideas. Feedback from the paper prototype aided our design of the interactive prototype. Technical

limitations building the interactive prototype had a significant influence on the final design, so we had to make many compromises from our original intent. The heuristic evaluation helped us identify where we had compromised too much on design, and we made modifications based on that feedback.

Testing a revised prototype with users in the usability test was the most valuable technique because it

showed us where our changes still fell short, and our final design takes these findings into consideration. Though each stage of the process informed our final design, the usability study gave us the most insight into the final design because users were using the real product on the phone. It gave us the chance to see how the interface would actually be used and informed us where our earlier design assumptions needed adaptation.

Final Interface Design

Overview

The last iteration of our application allows users to annotate a sample web page using Mobile Firefox and automatically saves annotations to a remote server for subsequent reference and review.

Features

Annotation Mode

In our current implementation, annotation is automatically switched "on" for the demo page; no special controls are used to activate it.

Help Animation

The first time the page is loaded, a splash screen appears with an animation demonstrating how to make a highlight using the interface (Figure 32). The help animation can be recalled at any time by tapping the '?' icon on the right side of the screen.

Highlight

A highlight is executed by single tapping the start of a block of text to highlight, then single tapping the end of a block of text to highlight. If a new highlight is created that is contiguous with an existing highlight, the two

highlights are automatically merged into a single highlight.

Starring

We implemented Gmail-like "starring" functionality. After a highlight is created a "hollow" star appears. If the user taps on the hollow star, it becomes filled and remains on the page. If the user instead takes any other action, the hollow star disappears. Stars persist in the interface, and show in Review Mode (Figures 33 & 34).

Final Interface Design (cont.)

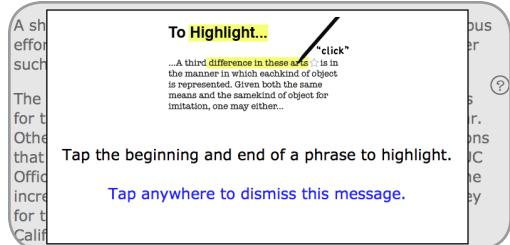


Figure 32

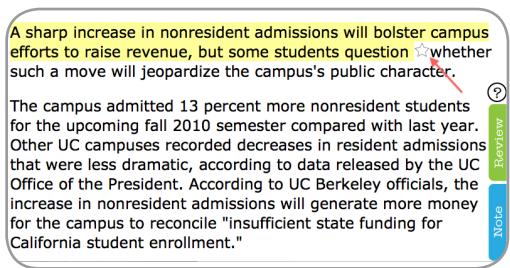


Figure 33

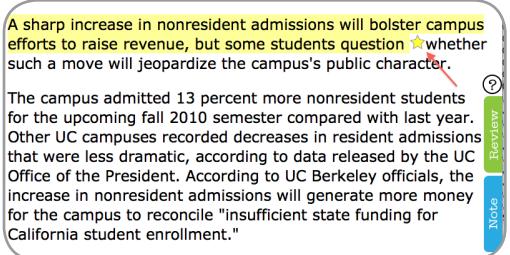


Figure 34

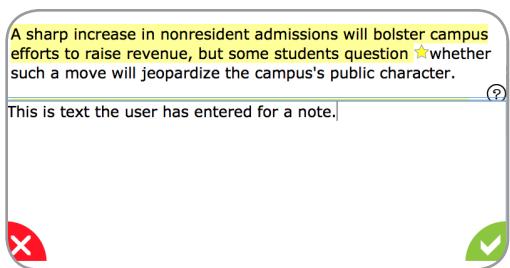


Figure 35

Notes

After a user makes a highlight, a “Note” control appears on the right side of the screen. Tapping this control opens a note entry box occupying half the screen. Text is entered using the physical keyboard on the N900. Once a note is finished, tapping the green check control saves the note, while tapping the red ‘X’ control cancels the note.

When a note is created, a speech bubble icon appears in the text after the last item highlighted. New notes are automatically tied to the last item that was highlighted.

Users can edit an existing note by tapping on the associated speech bubble icon. A note entry box is displayed, pre-filled with the previous content of the note. Users may make changes, and use the green check to save, or the red ‘X’ to discard changes.

The entry screen is optimized for a phone with a physical keyboard. The text entry box would be smaller on a phone that required a soft-keyboard (Figure 35).

A sharp increase in nonresident admissions will bolster campus efforts to raise revenue, but some students question whether such a move will jeopardize the campus's public character.

The campus admitted 13 percent more nonresident students for the upcoming fall 2010 semester compared with last year. Other UC campuses recorded decreases in resident admissions that were less dramatic, according to data released by the UC Office of the President. According to UC Berkeley officials, the increase in nonresident admissions will generate more money for the campus to reconcile "insufficient state funding for California student enrollment."

Review
Note

Figure 36

This figure shows the icon indicating a saved note. Tapping that icon allows a user to review and edit the note content (Figure 36).

Final Interface Design (cont.)

Review Mode

Once a user has made at least one annotation, a ‘Review’ control appears. Tapping the review control will cause the user to enter review mode.

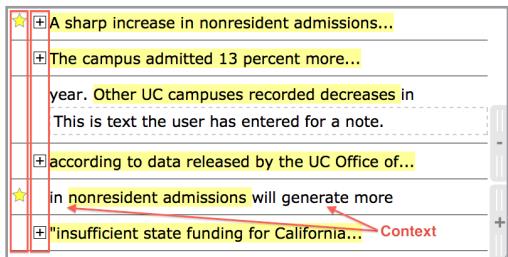


Figure 37

This image shows the dedicated columns for stars and plus-boxes, and a short highlight with context (Figure 37).

Highlight Snippets

All notes longer than one line are truncated to be one line in length. A trailing ‘...’ is appended to indicate that the note has been shortened.

Indicate Stars

Any highlights that were starred in the text appear as starred in the review interface. Stars have a reserved amount of space in the left-most column (highlighted in red).

Grouped Notes

Any notes associated with a highlight are displayed directly underneath that highlight, in a lighter font with a gray dashed border. Horizontal rules are placed after each highlight or highlight/note pair to indicate logical grouping.

Expand/Collapse Individual Snippet

A “plus box” control is automatically created next to any highlight that has been shortened to a snippet. Like stars, these controls have their own dedicated column. Tapping the control will expand the snippet to display the full text of the highlight. Tapping the control again will collapse the expanded text back into the snippet.

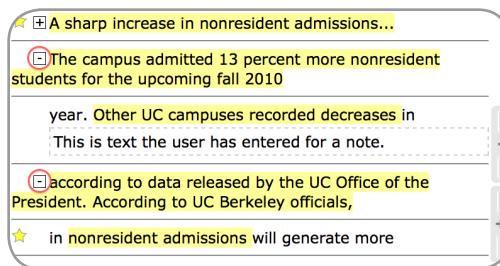


Figure 38

Figure 38 shows what an expanded highlight looks like. Note how the ellipses (...) from the previous image are converted to the complete text

for the highlights that have been expanded.

Expand/Collapse All Snippets

Two large “+” and “-” controls are in the lower right corner of the display. Tapping the “+” causes all snippets to show as expanded, regardless of their previous state. Tapping “-” causes all snippets to show as collapsed, regardless of their previous state.

Added Context

Any highlights that are shorter than one line are rendered with adjacent word(s) for context. The algorithm first adds the word immediately preceding the highlight. If additional space remains, the next word from the original text is appended. This process repeats until the addition of a new word would cause the length to exceed one line.

Return to Specific Highlights

Tapping on a highlight links the user to the original URL of the content and advances the page to the selected highlight. Users can also return to the original text via a URL link “Return to Text” rendered at the bottom of the review page (not pictured).

Final Interface Design (cont.)

Technical Implementation

Annotation Mode

Annotation Mode is implemented primarily in JavaScript, with the help of some Python scripts running under App Engine. Whenever the page is loaded, a server-side Python script checks to see if we have a record of an annotated version of the page in our database. If such a record exists, the annotated page is returned in the HTTP response. Otherwise, the default (non-annotated) page is returned. A client-side JavaScript function then places `` tags with unique IDs around each word in the text. These tags allow HTML-based annotations at the fidelity of single words.

Whenever a word is tapped, the mouse-up event is captured and the starting word ID is recorded. The next tap specifies the ending word. All words between the two words (inclusive) receive a highlighted CSS class and associated background color. A new empty star element is also created and placed at the end of the highlight. An “onclick” function allows the user to toggle the star as filled/unfilled, resulting in a class and image

change. Additionally, an html anchor tag is created keyed on the ID of the current word. This anchor tag allows the user to jump to specific notes when revisiting the page.

Tapping the note control causes a pre-programmed note-box to appear. Once a note is created and saved, a new note element is created and placed at the end of the associated highlight. Additionally, a hidden element is created with the content of the note. Whenever an already created note icon is clicked, the note-box is shown, and the inner HTML content of the note is set to the previously created content.

Performing any of the above annotation actions triggers an AJAX post message to our server’s API. The post contains the value of the current URL, and the complete content of the document’s body. The server receives the data and checks to see if there is already a record for the given URL in the database. It then adds or updates the information as appropriate.

When a user clicks on the ‘Review’ control, a hidden form containing the current URL is posted to the review page, which is then rendered.

Final Interface Design (cont.)

Review Mode

Most of review mode functionality is implemented server-side using Python in Google App Engine. The review page is passed the URL of the page to review when it is invoked from annotation mode. A server-side script performs the following actions:

- 1) Retrieve the complete body of the requested page from our application's database.
- 2) Read all words from the body into an array. Parse out all highlights, stars, and notes.
- 3) Construct highlight objects, with associated star and note properties.
- 4) Conduct operations on the highlights to format them for the page. Specifically, long highlights are given a "short form" property that contains text truncated to fit a single line. Short highlights are modified to include the surrounding words from the text.
- 5) Highlights are written out in order. Long highlight are written out with a hidden element containing the full text. Stars, notes, and "plus box" controls are added where appropriate.
- 6) The processed data is returned in the HTTP response.

Client-side JavaScript implements the "plus box" expand/contract functionality for long highlights. Additionally, "onclick" functions on each highlight item allow the user to tap on a note to return to the specified highlight on the original page.

Design

For annotation mode, our design is focused on being as unobtrusive as possible. We did implement a splash-screen tutorial that allows users to understand our somewhat unintuitive highlight controls. While we would have liked the highlight to be a swiping gesture, which users seem to easily understand, the Mobile Firefox interface captures all swiping motions for use in scrolling, making them effectively invisible to our script. We instead had to implement an idiom where the user taps the beginning and end of the section they want to highlight. While not immediately obvious, we found this highlighting method to be very learnable.

Stars and notes are added in-line with the highlight they are associated with. This close spatial proximity makes the relationship between the highlight and

note/star very clear, which is important on lines with more than one highlight.

Controls are rendered in the right margin next to the text, to avoid blocking words on the screen and to allow easy access from the right thumb. Because the controls are so narrow, we found that text labels were easier to comprehend than icons. We made the controls different colors to allow an experienced user to distinguish the functions without reading the text.

The one exception to our "keep it unobtrusive" design is the note creation dialog. As implemented the note text box takes up about half the screen when a user is typing a new note. We believe this divergence is appropriate as the user focus during this action is entering the text of their note. It is important that they are able to read everything they type, and are able to easily see and use the save/cancel controls. The "save" button is represented by a green check, which we again placed near the (usually) dominant right thumb. All users were immediately able to recognize that the green check was a "submit" action. We intentionally placed the cancel option on the opposite side of the note

Final Interface Design (cont.)

entry dialog, as far away from the save control as possible. Given the amount of effort it takes to input text on a mobile device, we wanted to make it difficult to accidentally lose work with a mis-tap. Again, the red X for cancel seems to be a well-understood idiom.

Review mode gave us more leeway to craft an interface suiting the needs of our users. Given that the purpose of reviewing is to see the annotations one has made, we placed a premium on maximizing the amount of annotation information on the screen. Long annotations are collapsed down to a single line, which can easily be expanded using the “plus box” located to the left of the annotation. By distilling highlights to single lines we were able to fit more individual highlights on a single screen. The plus boxes are optimized to be extremely responsive, minimizing the cost to the user of having highlights collapsed.

We did choose to sacrifice a small amount of screen real-estate on the left margin for holding the plus box but felt the plus box control is more than justified by the vertical space it saves by allowing us to collapse highlights. The controls for expanding/contracting

all highlights (located on the right next to the dominant thumb) are sufficiently narrow that they don't overlap text and utilize what would otherwise be dead space.

The star status indicator also was deemed sufficiently important to give it high prominence in the display. Our desire to preserve linear alignment for easy scanning led us to reserve space for a star/absence of a star in the left-most column. Since stars occupy this left-most position, an individual can very easily scan down their list of highlights to see what they designated as most important.

The final design choice we struggled with was how best to get users back to the original annotations. Currently, this is implemented by allowing users to tap any highlight to jump to that highlight in context. We found that users did not easily discover this functionality and could become trapped on the review page. We added an additional generic hyperlink as an escape hatch to allow users to return to the top of the original page. While this solution seems viable, it's possible that other subtle visual cues might better help users understand that they can tap a highlight to see

it in its original context. We decided that underlining and changing the color of the text to blue (as in standard hyperlinks) is visually too distracting but have considered adding subtle borders and visual depth to make the highlights look more like controls.

Limitations and Features Not Implemented

Not an Extension

Currently all code is implemented inside the demo pages on our server. As such, you can't use our application on any arbitrary page. As a result, it is currently more a proof of concept than a useful tool.

While we're unaware of any limitations that would prevent our application from being packaged in a Mobile Firefox extension, we were not able to do so in the time we had. We found the Mobile Firefox extension documentation very, very sparse, and ran into significant issues early in development that required significant research to surmount. We opted for a solution in technologies we understood to spend more time on the design and

Final Interface Design (cont.)

less time on the debugging.

As an alternative to an extension, it might be possible to create this application as a stand-alone tool if we gave users an easy way to import the pages they wanted to annotate into our application, then hosted this content.

Fragile

The HTML we use on our demo pages is very “clean” XHTML (valid XML). The parsers used to generate the review page are fragile, and break when given non-valid XML. To be truly useful our application would need much more robust error handling for dealing with the large body of malformed XML that exists on the web. Again, time constraints dictated that we invest more on the novel problems of our interface, instead of the less germane issue of cleaning up messy HTML.

Single User

Because we’re only running one demo page, we don’t require the concept of “Users” in our database. Adding the ability to create users and associating database entries with a specific URL-by-User pair would be essential to keep

users from overwriting each other’s annotations when they work on the same page. Supporting multiple users is relatively straightforward, but we opted not to given that our interface is really not ready for one user, much less multiple ones.

AJAX Limitations

Currently, all demo pages live on the same server. This greatly simplifies the use of AJAX calls for updating content. Cross-domain AJAX (as would be required for use in an extension) would require some significant rewrites of several functions.

Loading Stored Content

In the current demo, stored annotated content is loaded server-side before the demo page is rendered. If this application were running as an extension on a third-party site, it would instead need to replace the content of the remote page with the local stored copy. This can be accomplished with AJAX, but our application is not currently implemented in this way.

Deleting Highlights and Notes, Starring Existing Highlights

Our interface does not provide a way to “un-highlight” a piece of text. Similarly, there is no way to remove a note—deleting all the text does not delete the note icon. Also, if a user does not star a highlight at the moment it was created, they are unable to go back to the highlight later to star it. Ideally this functionality would be supported, and would be in the application if we had more time.

Desktop Interface

A lot of the power of our system would be in allowing users to work with their annotations on a desktop. The larger screen and richer input system could support much more advanced review operations, like searching, dragging and dropping quotes/citations for paper writing, tagging, and many other features we have not thought to explore. However, as the focus of the class was mobile interfaces, we made the mobile component the focus of our project, and leave the desktop interface as an area for future exploration.