I541 Project 2

Matt Snyder, Project Facilitator
Chad Camara
Thalith Nasir
CJ Page

Table of Contents

Design Rationale

"A seamless guide to content" – Project 2 mantra

Coming up with ideas for products such as this can be difficult because one has limited "real estate" to present the functionality to the users. In addition, this project was challenging because the problem space was not very specific. It was *definable* but not *quantifiable*. We thus defined the problem space (core) in terms of *context* and *disconnect.*

The *context* problem associated with RSS readers (Google Reader™) is that the RSS readers pull content from subscribed domains but do not show them in the context of the blog itself. By reading only the text from a website–not seeing it in context– means a loss of meaning and additional understanding from surrounding elements such as advertisements, color, pictures, comments, etc.

The *disconnect* problem becomes evident when users of such RSS readers leave the reader page itself to view the blog in its native locale. Leaving the reader to view the subscribed blog in its entirety or leaving a comment means becoming disconnected from your centralized blog feed. This too further complicated the contextual problem at hand and rendered the central locality of feeds useless.

We quickly found ourselves brainstorming ideas that offered more functionality, however they continued complicating the product. There was no feature or quick fix that could cure the users pain. To do this we had to not "fix" what we thought was wrong with current readers, but to look at what we thought our personas were looking for in an "RSS manager" tool. We wanted our tool to be simple, but did not try to shy away from building a powerful product. We organized our product's abilities within some simple rules of hierarchy.

While designing this product, we realized that every feature could not be presented to the user at once. Some abilities and features needed to be defined elsewhere, "off-stage" in the library of Firefox. Placing them there seemed appropriate and consistent with other features (ex. bookmarks), which had the same types of functions. Our design rationale includes the "off-stage" features and notes them in the descriptions below.

Toolbar position/size: We chose a thin toolbar because it blends in well with the rest of the browser, adding to the "seamless" portion of the design. It also takes up little real estate being positioned on top, because we don't want to inconvenience the reader by blocking browser content. Being positioned on top also lets the user easily activate/hide this tool when needed. We considered a sidebar, but decided that this would take up valuable screen space. Sidebars also tend to cut off the side of the webpage, making the user have to scroll left and right to see the content. A toolbar location at the top, when available to the user, only shrinks the length of the available browser window dedicated to content by a small amount. The navigation bar for our tool is presented in a way that allows the user to delve deeper into the content of blogs moving from left-to-right. *See appendix A.1, A.2*

Toolbar style: Our tool is consistent in color and look to the bookmark toolbar of the Firefox browser. This consistency again ties in with the mantra of keeping the design "seamless." The user sees UR Reader as a part of the browser, not as a separate entity. One does not have to "go to the reader," instead the reader stays with you. Future designs would incorporate well-thought-out icons differentiating the UR Reader bar from the bookmark bar.

Toolbar activation button: In addition to activating the Reader through the menus, we thought having a quick access button near the browser controls would be useful and convenient. This button has ability to hide/show the UReader tool bar at anytime. Another reason for locating the reader button near the address bar, was so that user could quickly subscribe to a blog by dragging the current URL over to the reader button in the same way you can set your homepage. *See appendix A.4*

Blog auto-detection: If the user navigates to a subscribed site without using the UReader, the reader toolbar will automatically be displayed. This feature helps the user by reminding them that this tool is available to guide them. This feature can be turned off in a Firefox preference pane.

Subscriptions: A user can subscribe and organize subscriptions through Firefox's library, which already manages bookmarks and history. By piggybacking some of Firefox's functionality, we avoided requiring the user to learn something

new. Using Firefox's subscription organization tools keeps the toolbar clean and lets the user focus on navigation and content; it doesn't distract the user with blog management.

Adding subscriptions: By dragging the URL over to the RSS activation button, one can add the site to his/her blog subscriptions. This is an advanced feature and we don't expect users to recognize this on day one. This short cut is very useful because it allows the user to quickly subscribe without having to go through the browser menus. In the interest of keeping the reader bar simple and clean, we didn't put an option there letting the user "add" a subscription. There were already three different ways to subscribe to a RSS feed: the menu bar, dragging the URL to the RSS button, and the subscription options within the site itself.

Subscription flow: The user accesses/visits subscribed feeds by 1) looking up subscriptions, and then 2) selecting displayed posts from the selected subscription. This one-two process helps users quickly navigate from one post/blog to the next without disconnecting the user from the content.

Subscription button: This is a button that shows a drop down menu listing current subscriptions. This is the first step a user takes in accessing their subscribed content. It is a drop down menu to avoid blocking content and it is in list form to be scalable and easy to manage. By default it shows the user's top 10 most read subscriptions. The organization and number of subscriptions shown can be changed in preferences.

Site name button: Once a subscription has been chosen to navigate to, this drop down menu shows a list of new and old posts of the currently selected subscription. This button is a separate button (step 2) from the subscription button because it gives the user granular control over selection. It further promotes the idea that the reader is a guide, and as a guide it helps to narrow the user's focus to specific content.

New/old blog status: The reader has the ability to track what posts the user has read or not read. We also have given the user the ability to mark posts manually as "read" or "unread." We haven't fully developed how this feature works, or if the labels "read" and "unread" are ideal, but we have agreed that is an important aspect of the final design.

"Next new post" button: This feature allows users like Stacey (persona 1) to quickly go through a site, viewing one post to the next rapidly. This is good for viewing sites with little textual content or sites where all posts are relevant. Stacey is concerned with the experience of visiting a site as a summation of all the posts.

Comments button: This button shows activity on something the user commented on. Cliff, our second persona, puts a lot of thought into the comments that he posts. When someone responds to his comments, he wants to know immediately because the conversation is important to him. The idea here is that in order to keep the comments in context, the user would want to see conversation directed at the user's comments. "Activity" is defined as a direct response to the user's comment. Specifically this is any direct response as allowed by particular sites, or any response that includes the user's name or quotes the user. Additional criteria could also be used if deemed to directly relate to the user's original comment..

Search blogs/posts: In order for the reader to be a "guide to content" and provide context, the user would need to be able to search through previously read posts. It is assumed that context for the user consists of the connection between new and old information. The search is integrated into the main web search function of the browser. The RSS search option becomes one of the list of options already part of the search field. The alternative that was considered was to add a search field on the UR Reader toolbar. This was discarded in order to keep the number of controls on the bar to a minimum, and to take advantage of current browser functionality.*See appendix A.5*

Personas

About our personas: Our personas attempt to represent 2 distinct blog readers ranging in age from 24 to 45. We built our personas based on "active" and "passive" roles. These two contrary persona types represent both ends of the online-content-contribution spectrum. "Active role," meaning that one contributes to the online community at hand, regularly posting and reviewing their own responses to blog posts–feeling part of a community and connecting to others around the world who have similar interests. "Passive role", meaning that the Internet and blogs are primarily used as a source of information, and one only contributes to content if he/she feels like it will have value to someone else reading the blog. A passive persona does not feel a need to belong to an online community in the same sense as an active persona.

Stacey
24-year old GAP employee with a BA in Fashion Design. She reads and posts frequently. (Active)

- Entertainment
- Friends
- Fashion
- Social

Stacey is a 24-year old recent college graduate, with a degree in fashion design. She just became a manager at the GAP. She loves her job because she loves clothes and interacting with people.

She uses the Internet for looking up job postings, reviewing fashion news, and staying current with popular entertainment. She is motivated to interact with content and blogs because it keeps her up to date with trends and gossip in the fashion world. She contributes to legitimize her presence in online communities.

Cliff
45-year old Advertising Exec.
Reads blogs a lot and posts little (Passive)

- Business sites
- Stock trading
- News/stock ticker
- Sports

Cliff is a 45-year old CFO for a local advertising agency in Chicago. He loves sports and crunching numbers.

He uses the Internet for looking up financial information and business trends relating to the advertising industry as well as staying up-to-date on sports. Cliff posts a lot on fantasy football because he likes to talk trash with his friends on line. He peruses forums and business matters but generally doesn't participate in conversations on line. Although he regularly maintains and updates his own blog, he rarely participates and comments on other blogs. When Cliff does post however, he is very interested in the conversation he engages in.

Testing procedures

In this project, we tried using *low-fi* mockups as described in Bill Buxton's book *Sketching* on page 371-391. We used a slightly modified screen shot of Firefox, adding a button and a tool-bar. We then used hand-written buttons in our tool-bar and Post-It notes during testing to simulate interface interaction.

Testing with low-fi mockups in this situation was very useful and efficient. The lack of detail in the interface also was a good alternative to the sometimes-distracting nature of high-fi mockups. The user was more focused on the interactive nature and process instead of specific colors and shapes.

After testing our first round with users, it became obvious that in order to further engage the user, we needed to better define the shapes of buttons and drop down lists, by producing more refined mock-ups.

NOTE: We decided to start our test with a lead in. These lead in questions were used to assess someone's role (active vs. passive) in contributing to online content and blogs. Because this product is based on RSS, we needed to know how much of an understanding the user had about RSS. We felt like if they understood a lot about RSS, we would refer to our product as a "new RSS reader manager". If they blogged but didn't know about RSS, we would refer to our product as a "Blog manager". We thought there was value to referring to our product in whatever manner appropriate as to not alienate the user simply because of an unknown term.

Pre-test questions:

1. Do you know what RSS is?
2. Do read, write, and visit blogs?

Lead in statement: "This is your first time using this improved version of Firefox, where would you look to access the blog/RSS manager?"

Test Questions

1. You have been using this new tool for a couple of months now, how would you look up your favorite blog, "BoingBoing?"
2. How would you find the unread post titled "Skeleton Letterpress Bookplates?"
3. Let's say you wanted to know more about the post "The Elsberry Paradox" before going there and you hover over the title, and this tool-top pops up, would it be useful?
4. You want to take a break from technology news and check out your next favorite blog, "Icanhascheezburger" – please pull up that blog using our tool.
5. In this blog you don't care about sorting through specific posts, you just want to view all the unread posts since your last visit, how would you go about doing that?
6. Would you find the “next new post” button/feature useful?
7. Is there anyway this button/feature could be displayed so that its usefulness is more obvious?
8. You left a comment on www.boingboing.net a couple of hours ago. How would you find out if there has been any other activity on the same post?

Conclusion: Ask improvements, likes/dislikes

User Test Results

User testing yielded many positive results, but a few problem areas arose that are addressed in the redesign. It is pertinent to note that we do not take the positive trend in the results as absolute affirmation of our design. We are encouraged to continue with certain design aspects because of it, but also aware that different questions and methods of testing should be incorporated to better identify the good and the bad of the design.

Positive Feedback

The positive feedback that led us to keep with certain design aspects included:

- The look and feel was “expected,” and seemed to be a natural fit for the Mozilla browser
- The list of subscriptions was easy to find
- It was easy to understand the flow of the left to right lists in order to access deeper content
- Keeping the functions “relevant” was appreciated
- The drop-down list of posts made it easy to navigate to specific posts

Negative Feedback and Redesign

The problem areas came from behavior that was unanticipated as well as small things that were overlooked in the initial design. User 1 was surprised to see that the subscription list didn’t expand to show the posts from the currently highlighted subscription. He felt that it wasn’t natural or convenient to click the subscription and then have to navigate to another list to choose the post. This led us to add the expanding list to the right of the subscription list. We added this functionality, but did not eliminate the list of posts under the site name. This also made us realize that when the toolbar initially drops down, having “Site Name” and “Next New Post” is not relevant until a subscription has been chosen. This led to the decision to only show the “Subscriptions” and “My Comments” buttons at first. Then once the subscription has been chosen, “My Comments” slides to the right and makes room for the “Site Name” and “Next New Post” buttons.

During the initial design phase, it was difficult for us to decide how important search functions would be. However, when User 3 was asked to start the reader or to go to a subscription, his natural tendency was to go to search. User 3 saw the search function as his “guide.” While this behavior was not anticipated and might be categorized as “outlying,” it did help us to decide that including the RSS Search into Mozilla’s search was indeed necessary, and might need to include additional search functions for this type of user.

An issue that was not clearly addressed in the first 3 user tests was how to subscribe to an RSS feed. Two short user tests were conducted to see how users would expect to subscribe other than using the subscription options on the site itself. The first user was not shown the menu bar across the top of the screen, as would be seen on the top strip of the desktop on a

Macintosh OS, or as the top strip of the browser window on a Windows OS.

This first user tested could not identify at all how a subscription would be added. The second user was shown the menu bar with an additional RSS menu option as the last option on the right. Before even bothering with how to subscribe, his first question was why we didn't call "Subscriptions" "My Subscriptions." This was an observation that we somehow had overlooked and decided to implement in the redesign. Then when he looked for subscriptions, he immediately recognized that he could add subscriptions by going to the RSS Subscription options on the menu bar at the top. He also asked if he could add from the subscription list, which was currently not a function we had thought of and subsequently was incorporated in the redesign.

We also asked that *if* he knew that he could subscribe by dragging the address onto the RSS button, would he use it. He was very interested to know this and said that he would definitely use it, and also would consider it helpful to subscribe by dragging the address to the subscriptions button on the toolbar. This was an additional function that was added in the redesign.

Post-test design rationale

Added expanding menu: Shows posts when hovering on the subscriptions lists. This eliminated the need to click a subscription prior to being able to see the posts. *See appendix C.1 and C.2*

Eliminated unnecessary buttons: "Site Name" and "Next New Post" were deemed unnecessary when the toolbar first shows up. If the user has not selected a subscription, then there would be no site name and no next post to navigate to. Instead, the buttons appear when the user chooses a particular subscription.

Included RSS search capability: The debate over whether or not this was necessary was ended due to the positive feedback

from users, as well as certain user tendencies to rely on search as the guide.

Labeling: Changed "Subscriptions" to "My Subscriptions." It is "your reader," not just anybody's reader, so it follows that the label should state this difference.

Subscribing from main list: Added an option at the bottom of the "My Subscriptions" list that enables the user to add subscriptions directly from the list.

Drag and drop subscribing: Added the ability to subscribe to a feed by dragging the address into the "My Subscriptions" list. This might be more natural for the user than dragging into the button. However, that ability was not omitted and would be easier given that the user would have to click the RSS button in order to have a "My Subscription" to subscribe to.

iPhone design rationale

Designing for the iPhone poses several new challenges as well as opportunities that the desktop based browser does not. One such challenge is adapting to the small form factor of the iPhone interface. An opportunity, on the other hand, is the fact that users would be carrying our application with them all the time.

We felt the flow of our Firefox design lends itself to the iPhone quite well with the touch interface being very conducive to a linear left-right flow of menus. One of our personas, Cliff, formed the basis of our design decisions when we began sketching the iPhone version of UReader. Keeping the core in mind, the UReader iPhone app displays the blog post as it is displayed on the actual website. To aid in reading, the app would orient itself if the user swivels or rotates the iPhone to 'landscape' mode. This behavior is also consistent with the behavior of other iPhone apps, specifically the Safari browser.

Personal experience tells us that the iPhone is great for reading however troublesome to enter text with. So we revolved around the idea of making it easier to read and track

comments. We're assuming the user would comment on their computer so they can sync the iPhone with UReader. Currently, the desktop version of UReader does not allow syncing, as this feature was deemed important *after* the design of the iPhone version. These assumptions and our personas informed the basis of design decisions detailed below.

Overall style/colors: The overall look and feel of the application is meant to be consistent with the look and feel of the iPhone interface. Users already familiar with how things work on the iPhone would not have to relearn or acquaint themselves to a differently styled interface.

Subscriptions button: Similar to UReader in Firefox, the iPhone design includes a "My Subscriptions" button which allows users to view blogs they are currently subscribed to. The total number of unread posts is also indicated within parentheses. *See Appendix D.4, D.5.*

Comments button: This takes the user to the My Comments section where they can see comments that were directly in response to the user's comment. This feature is especially helpful to our persona, Cliff who would use it to track responses to his comments while on the go.

Follow up status: Considering the different problem space, we realized that Cliff would tend to avoid using the iPhone interface to type whenever possible. Personal experiences have shown that while the iPhone may be useful to read text, typing on it is a rather difficult task for most users. Cliff would expect to write deeper, more thoughtful posts than Stacey and thus he would rather avoid extended periods of time typing on the phone. The 'Follow Up' feature would allow Cliff to tag items, which allows him to come back to specific posts. The UReader app on the iPhone can also be synced with the browser version ensuring data consistency.

Navigation buttons/location: Consistent with current iPhone navigation positioning, the iPhone UReader positions the Back button, Follow Up star and Next Post button at the top of the screen. We had considered positioning the Follow Up star and Next Post button at the bottom but decided to save real estate considering the fact that the actual website is being displayed on screen. A button to take the user back to the first screen (the menu) was also considered. However, it is rather easy to

flow back to previous screens by touching the top left button, which is also consistent with the iPod application in the iPhone. As a result, to reduce clutter, we decided against having a dedicated 'Home' button.

Next New Post button: This button takes the user to the next unread blog post. This is displayed in the form of an icon (a right arrow) to conserve screen real estate. The icon is also consistent with what is used in the iPhone interface to indicate forward flow. *See Appendix D.7*

Star button: When a user comes across a post that requires commenting later (at a computer), or rereading, clicking the star button on the menu would add the post to the Follow Up item list in the home menu. *See Appendix D.7*

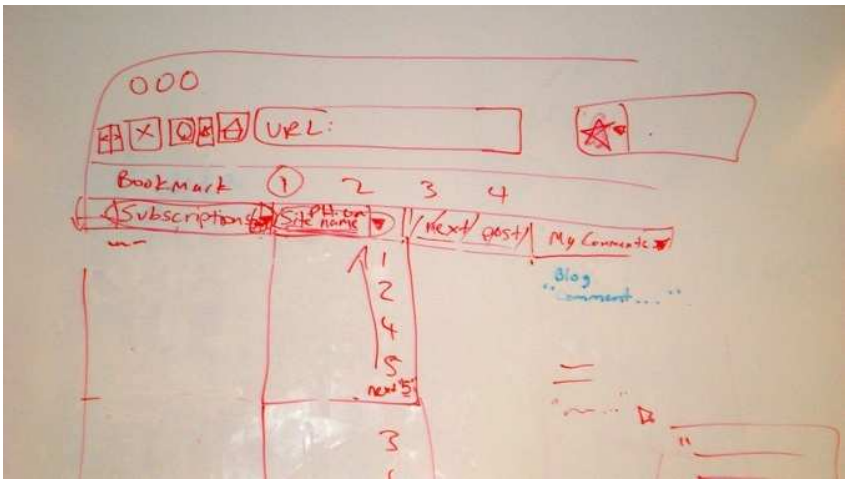
Home Screen Icon: The icon for our iPhone app is styled on the basis of the RSS icon. Similar to the Mail and SMS applications on the iPhone, the UReader icon displays the number of new/unread posts in a red circle. *See Appendix D.8*

Alerts: Similar to how a new SMS is displayed; the user can be alerted if there has been activity on any of the user's comments. This is something Cliff would use since he has relatively more 'meaningful' feedback on comments. This implies that the quantity and frequency of comments is something quite manageable. Stacey, on the contrary, would be flooded with new comments in a matter of moments due to the nature of sites like PerezHilton.com and Icanhazcheezburger, where comments are very frequent. As a result, this feature would be off by default to accommodate the majority of users who would find this feature annoying, but giving the option of users like Cliff, to turn it on.

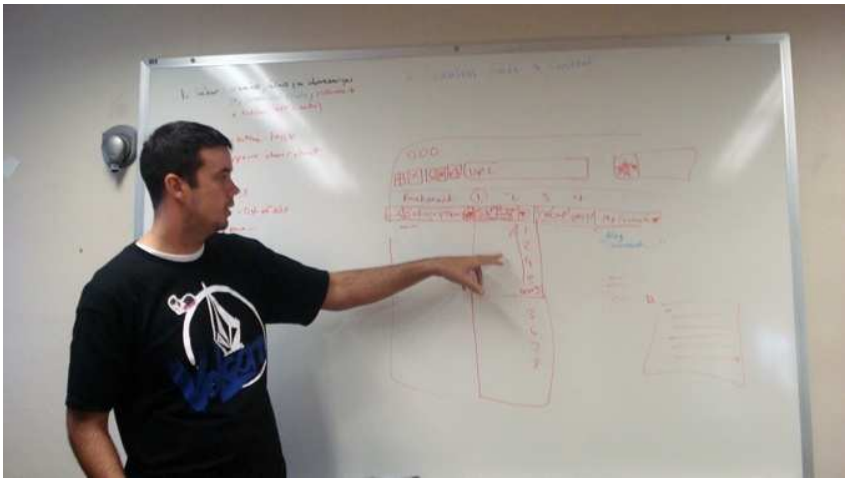
Appendix

Appendix A – Sketches

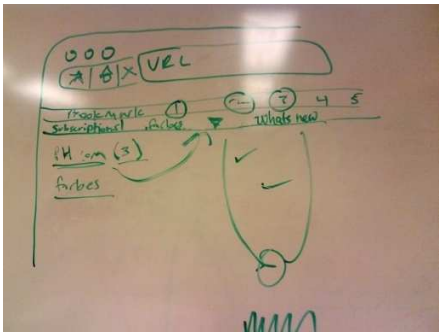
A.1



A.2



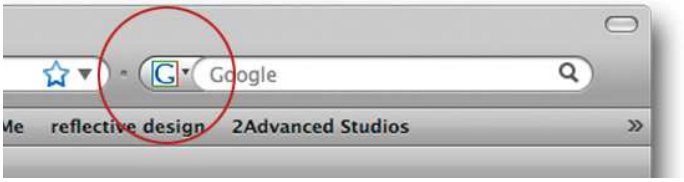
A.3



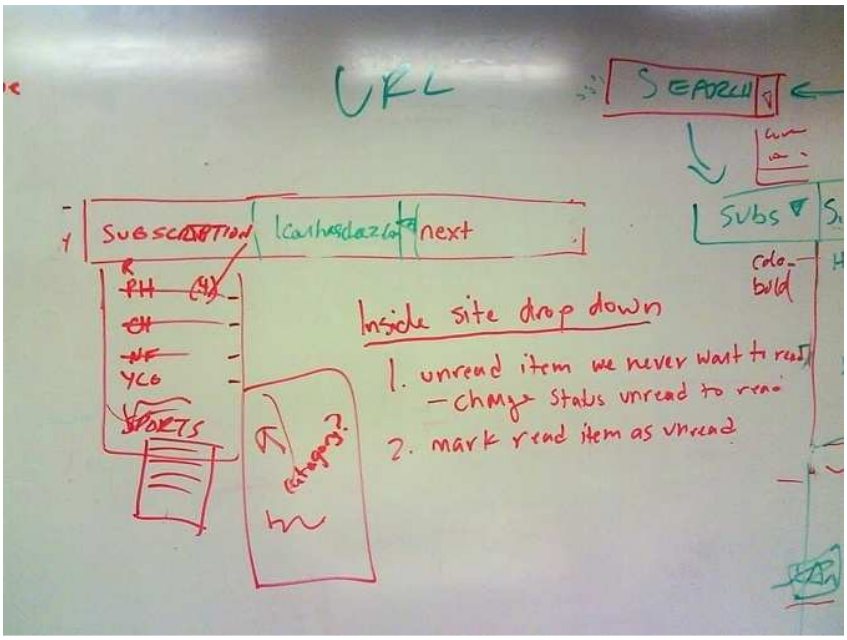
A.4


Most Visited
Gmail
OneStart
Dan's Blog
idp08 purple

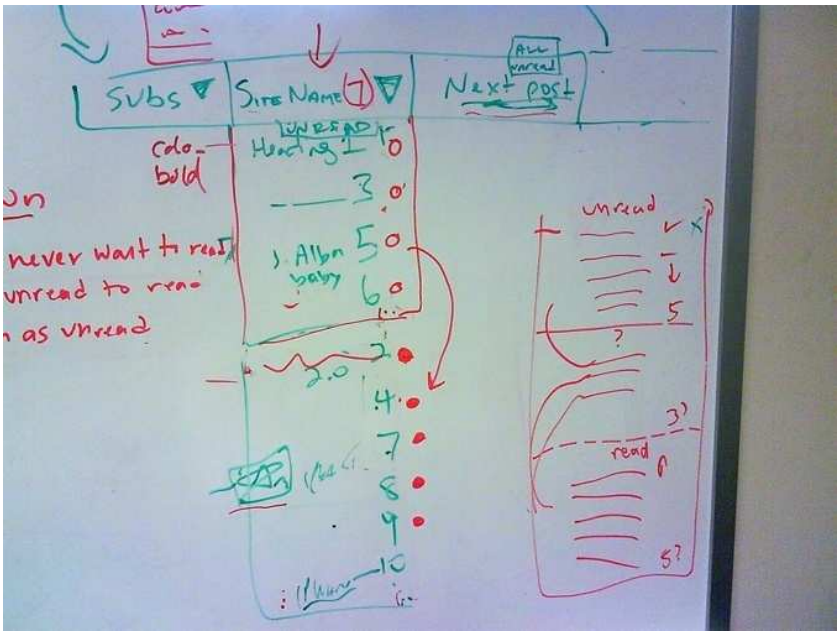
A.5


Google
reflective design
2Advanced Studios

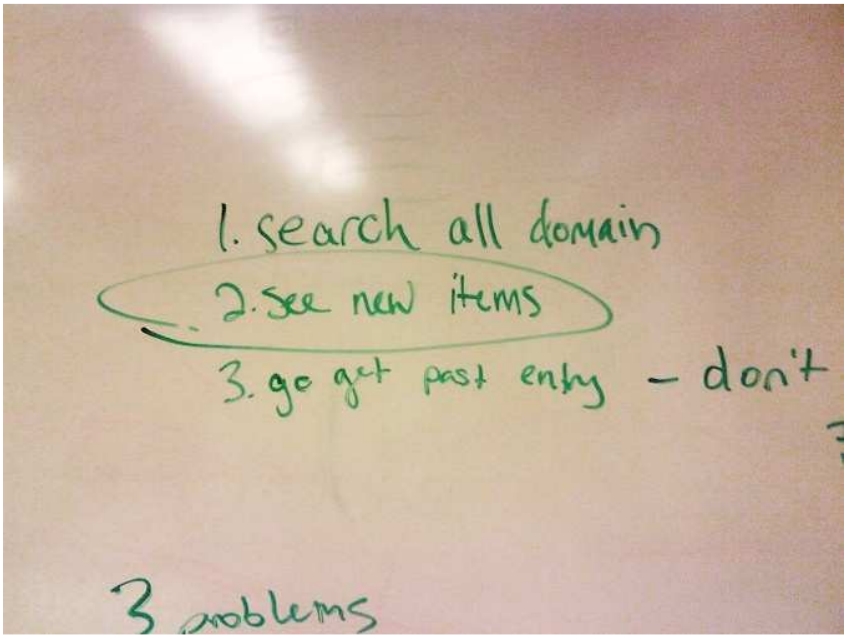
A.6


URL
SEARCH
SUBS
next
Inside site drop down
1. unread item we never want to read
2. mark read item as unread

A.7



A.8



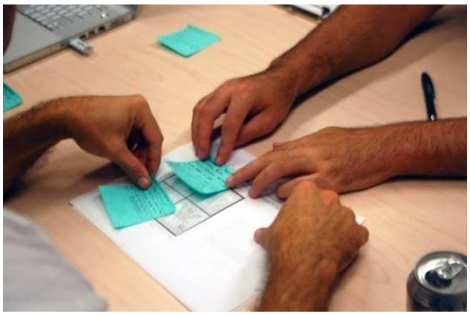
All sketches can be found on supplemental CD, *Appendix A*

Appendix B – paper prototype

B.1

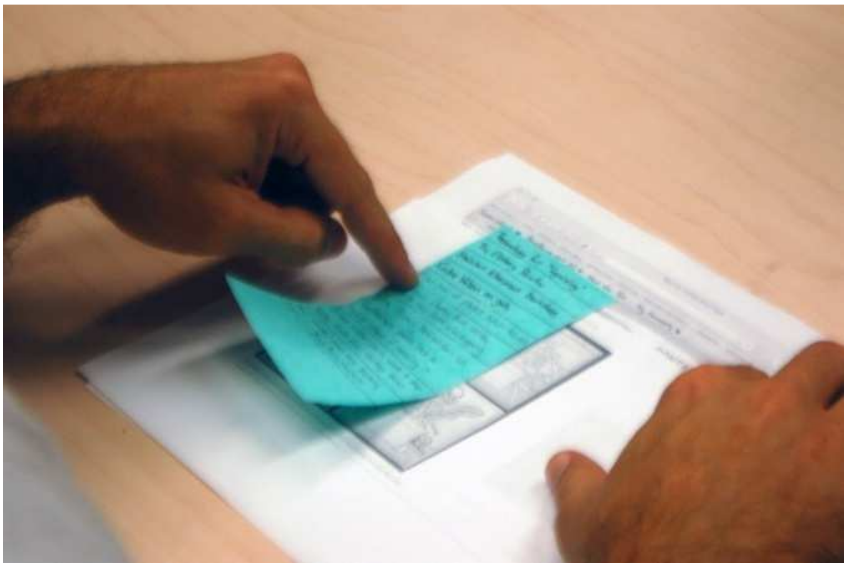
B.2

B.3

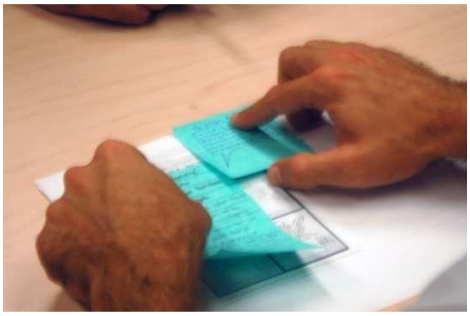


B.4

B.5



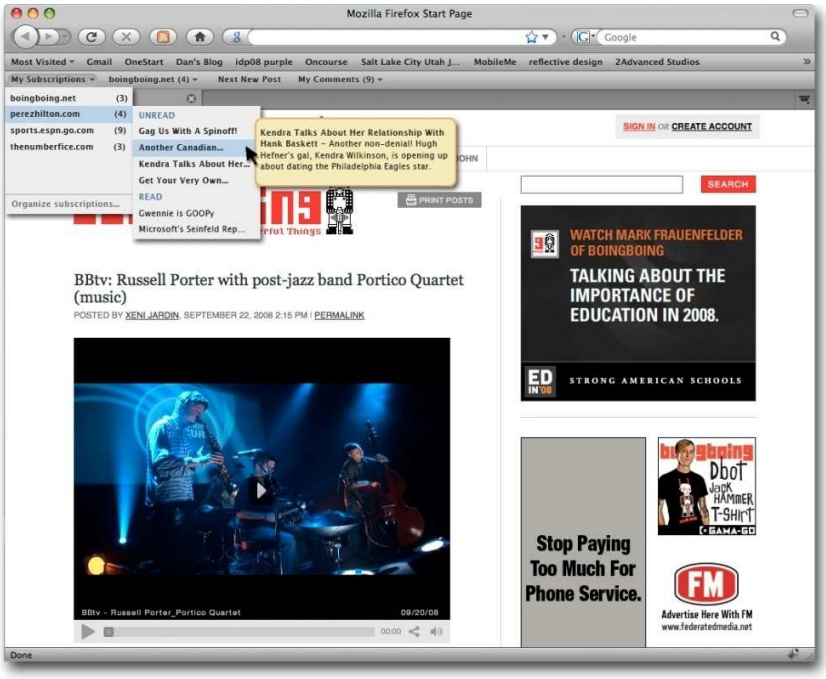
B.6



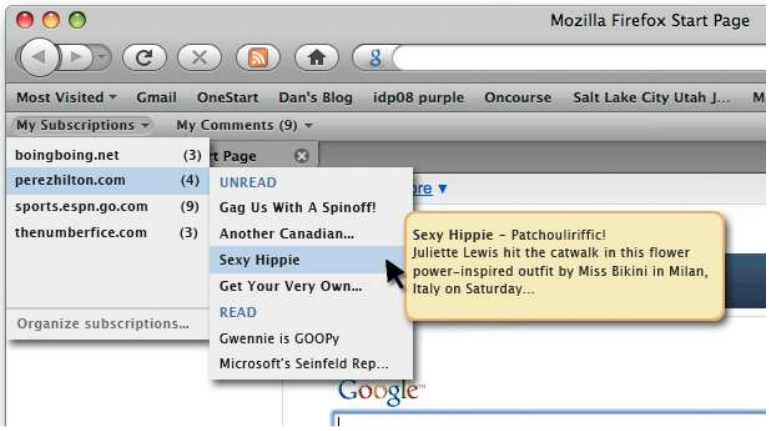
All prototype images can be found on CD, *Appendix B*

Appendix C – post-test sketches and design

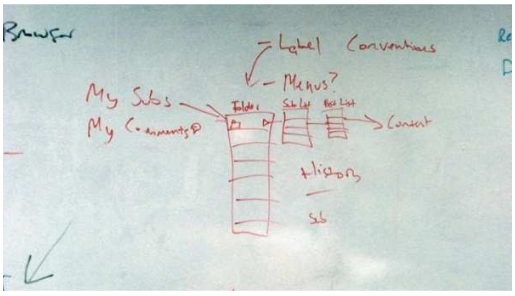
C.1



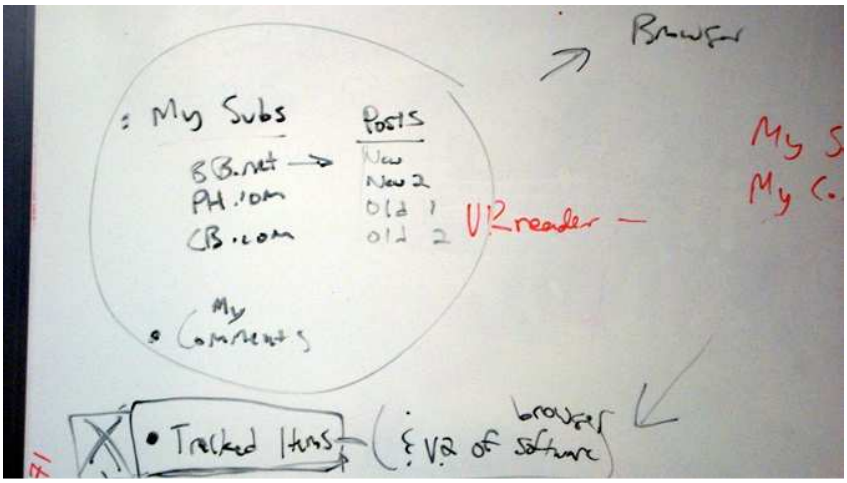
C.2



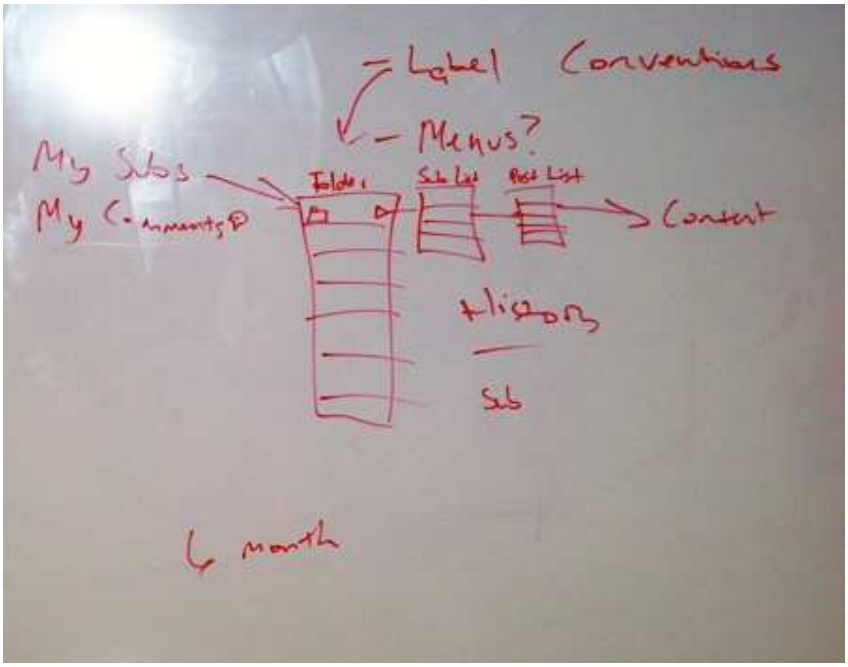
C.3



C.4



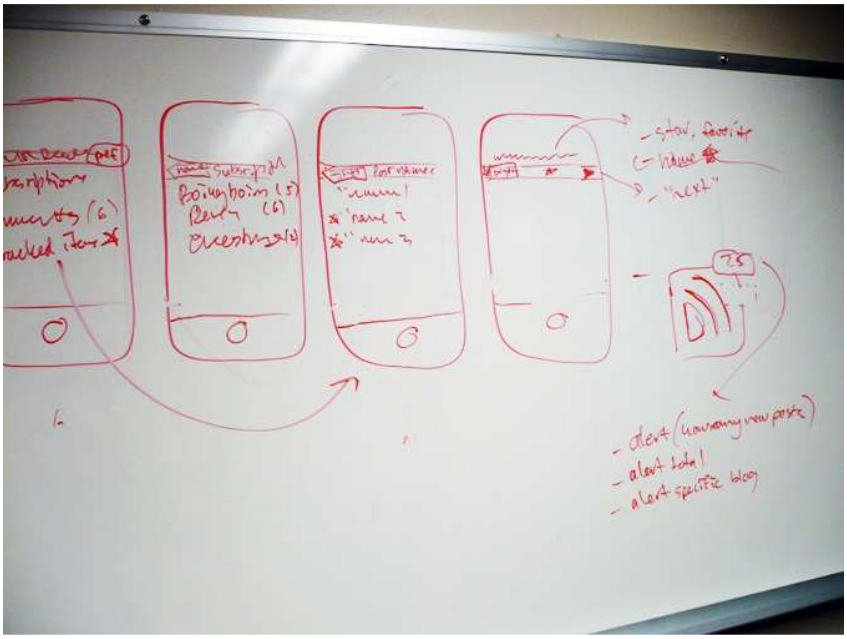
C.5



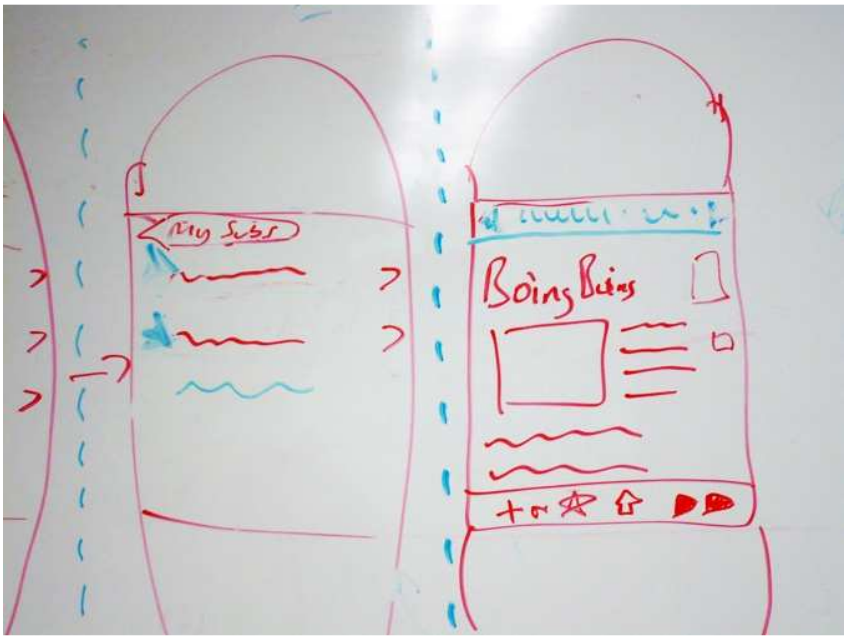
All prototype images can be found on CD, *Appendix C*

Appendix D – iPhone sketches

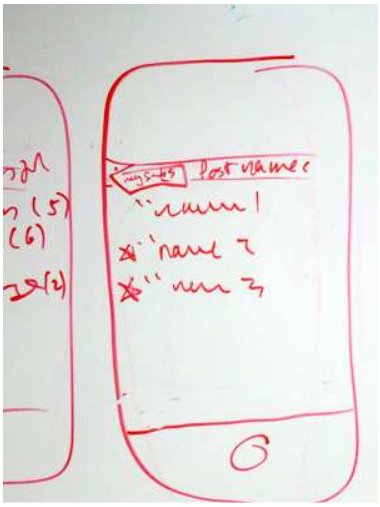
D.1



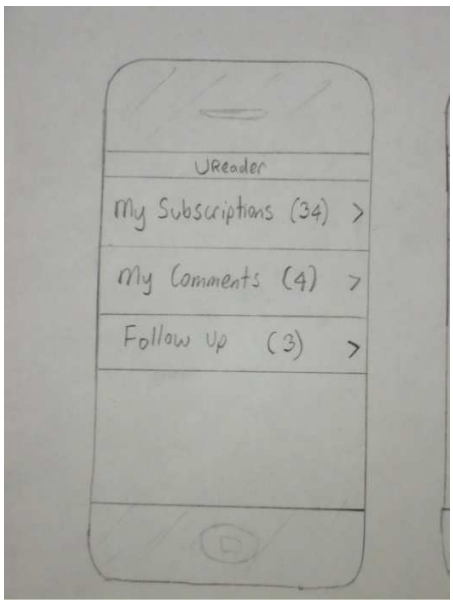
D.2



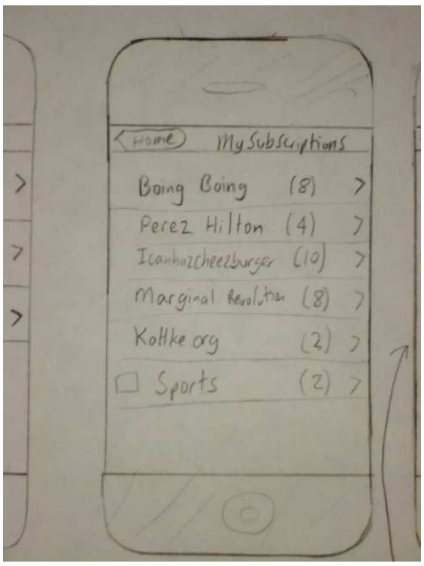
D.3



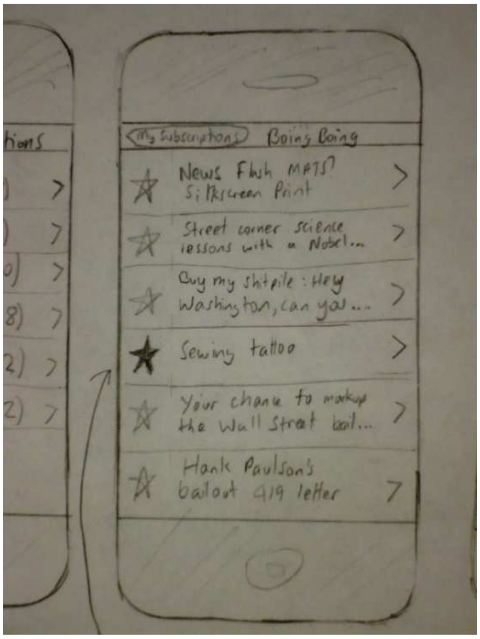
D.4



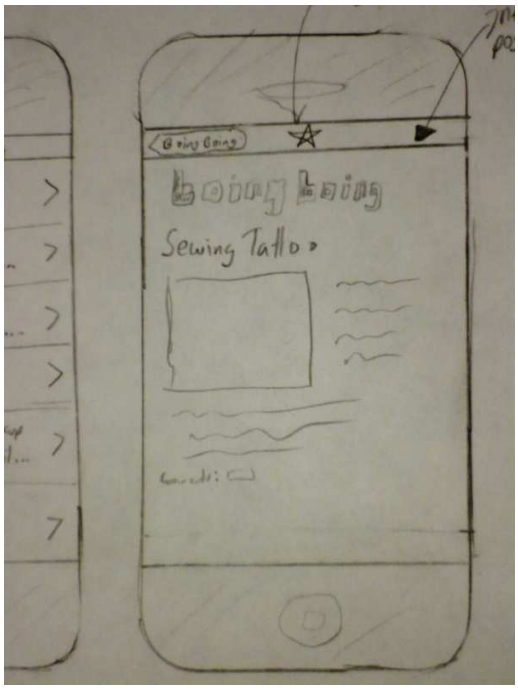
D.5


Home
My Subscriptions
Boing Boing (8)
Perez Hilton (4)
Marginal Revolution (8)
Kottke.org (2)
Sports (2)

D.6


My Subscriptions
Boing Boing
News Flash MAPS? Silkscreen Print
Street corner science lessons with a Nobel...
Buy my shitpile: Hey Washington, can you...
Sewing tattoo
Your chance to make the Wall Strat bail...
Hank Paulson's bailout 419 letter

D.7



D.8

All prototype images can be found on CD, *Appendix D*